**Desenvolvimento de técnicas de aprendizado de máquina via
sistemas dinâmicos coletivos**

Roberto Alves Gueleri

Tese de Doutorado do Programa de Pós-Graduação em Ciências de
Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Roberto Alves Gueleri

Desenvolvimento de técnicas de aprendizado de máquina via sistemas dinâmicos coletivos

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Zhao Liang

USP – São Carlos
Setembro de 2017

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

G925d Gueleri, Roberto Alves
Desenvolvimento de técnicas de aprendizado de
máquina via sistemas dinâmicos coletivos / Roberto
Alves Gueleri; orientador Liang Zhao. -- São
Carlos, 2017.
146 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, 2017.

1. Aprendizado computacional. 2. Computação
bioinspirada. 3. Redes complexas. 4. Processamento
de imagens. 5. Comportamento coletivo. I. Zhao,
Liang, orient. II. Título.

Roberto Alves Gueleri

**Development of machine-learning techniques via collective
dynamical systems**

Doctoral dissertation submitted to the Institute of
Mathematics and Computer Sciences – 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. Zhao Liang

USP – São Carlos
September 2017

AGRADECIMENTOS

Agradeço à minha família, por me ajudar e me apoiar em todas as decisões que tomei ao longo da vida, inclusive na decisão de realizar esta tese de doutorado. Agradeço ao Prof. Dr. Zhao Liang, orientador no desenvolvimento desta tese, pela paciência e dedicação que teve ao longo desses anos. Mais que um orientador, ele se tornou um grande amigo. Agradeço aos demais professores e pesquisadores, do ICMC-USP e de outras instituições, do doutorado e de outras épocas, que de alguma forma contribuíram para o desenvolvimento desta tese e para a minha formação. Agradeço a todos os amigos que estiveram comigo nesses anos, que não só me ajudaram profissionalmente, mas permitiram também ótimos momentos de descontração. Enfim, expresso aqui minha gratidão a todas as pessoas que, direta ou indiretamente, contribuíram para mais esta etapa que agora estou completando. Felizmente tive saúde, motivação, energia, oportunidades e circunstâncias favoráveis para conseguir concluir todo este trabalho, embora alguns momentos difíceis e noites mal-dormidas também façam parte. Muito importante também foi o apoio financeiro concedido pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), por meio de bolsa (Processo 2013/08666-8), que permitiu minha dedicação exclusiva ao desenvolvimento desta tese de doutorado.

RESUMO

GUELERI, R. A. **Desenvolvimento de técnicas de aprendizado de máquina via sistemas dinâmicos coletivos**. 2017. 146 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

O aprendizado de máquina consiste em conceitos e técnicas que permitem aos computadores melhorar seu desempenho com a experiência, ou em outras palavras, “aprender” com dados. Duas de suas principais categorias são o aprendizado não-supervisionado e o semissupervisionado, que respectivamente consistem em inferir padrões em bases cujos dados não têm rótulo (classe) e classificar dados em bases parcialmente rotuladas. Embora muito estudado, trata-se de um campo repleto de desafios e com muitos tópicos abertos. Sistemas dinâmicos coletivos, por sua vez, são sistemas constituídos por muitos indivíduos, cada qual um sistema dinâmico por si só, de modo que todos eles agem coletivamente, ou seja, a ação de cada indivíduo é influenciada pela ação dos vizinhos. Uma característica notável desses sistemas é que padrões globais podem surgir espontaneamente das interações locais entre os indivíduos, fenômeno conhecido como “emergência”. Os desafios intrínsecos e a relevância do tema vêm motivando sua pesquisa em diversos ramos da ciência e da engenharia. Este trabalho de doutorado consiste no desenvolvimento e análise de modelos dinâmicos coletivos para o aprendizado de máquina, especificamente suas categorias não-supervisionada e semissupervisionada. As tarefas de segmentação de imagens e de detecção de comunidades em redes, que de certo modo podem ser entendidas como tarefas do aprendizado de máquina, são também abordadas. Em especial, desenvolvem-se modelos nos quais a movimentação dos objetos é determinada pela localização e velocidade de seus vizinhos. O sistema dinâmico assim modelado é então conduzido a um estado cujo padrão formado por seus indivíduos realça padrões subjacentes do conjunto de dados. Devido ao seu caráter auto-organizável, os modelos aqui desenvolvidos são robustos e as informações geradas durante o processo (valores das variáveis do sistema) são ricas e podem, por exemplo, revelar características para realizar *soft labeling* e determinar classes sobrepostas.

Palavras-chave: Aprendizado de máquina, comportamento coletivo, movimento coletivo, flocking, sistemas auto-organizáveis .

ABSTRACT

GUELERI, R. A. **Development of machine-learning techniques via collective dynamical systems**. 2017. 146 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

Machine learning consists of concepts and techniques that enable computers to improve their performance with experience, i.e., “learn” from data. Unsupervised and semi-supervised learning are important categories of machine learning, which respectively consists of inferring patterns in datasets whose data have no label (class) and classifying data in partially-labeled datasets. Although intensively studied, machine learning is still a field full of challenges and with many open topics. Collective dynamical systems, in turn, are systems made of a large group of individuals, each one a dynamical system by itself, such that all of them behave collectively, i.e., the action of each individual is influenced by the action of its neighbors. A remarkable feature of those systems is that global patterns may spontaneously emerge from the local interactions among individuals, a phenomenon known as “emergence”. Their relevance and intrinsic challenges motivate research in various branches of science and engineering. In this doctorate research, we develop and analyze collective dynamical models for their usage in machine-learning tasks, specifically unsupervised and semi-supervised ones. Image segmentation and network community detection are also addressed, as they are related to machine learning as well. In particular, we propose to work on models in which the objects’ motion is determined by the location and velocity of their neighbors. By doing so, the dynamical system reaches a configuration in which the patterns developed by the set of individuals highlight underlying patterns of the dataset. Due to their self-organizing nature, it is also expected that the models can be robust and the information generated during the process (values of the system variables) can be rich and reveal, for example, features to perform soft labeling and determine overlapping classes.

Keywords: Machine learning, collective behavior, collective motion, flocking, self-organizing systems .

LISTA DE ILUSTRAÇÕES

Figura 1 – Velocidade das partículas após algum tempo de evolução, para diferentes valores de densidade e ruído	29
Figura 2 – Valor absoluto da média normalizada das velocidades (v_a) em função do nível de ruído η , para diferentes quantidades de partículas N	30
Figura 3 – As três leis de interação do modelo de <i>boids</i>	30
Figura 4 – Conjunto artificial de dados, cada qual caracterizado por dois atributos	33
Figura 5 – Dois exemplos de kernel locais	37
Figura 6 – Acurácia de rotulação em função dos parâmetros λ e Δ , sobre uma rede artificial com 1000 vértices, distribuídos em quatro classes de igual tamanho	40
Figura 7 – Ilustração do processo de classificação via competição de partículas	41
Figura 8 – Resultados de classificação sobre dois conjuntos artificiais de dados	41
Figura 9 – O comportamento da função $f(d_{ij}) = d_{ij} \cdot e^{-\beta d_{ij}^2}$, para $\beta = 1$	42
Figura 10 – Conjunto artificial de dados, formado por duas gaussianas contendo 400 objetos cada, 20 deles previamente rotulados.	43
Figura 11 – Comportamento dinâmico da técnica proposta, sobre o conjunto da Figura 10, empregando $\beta = 1$ e $p = 0,01$	44
Figura 12 – Comportamento dinâmico da técnica proposta, sobre o conjunto da Figura 10, empregando $\beta = 50$ e $p = 0,01$	45
Figura 13 – Ilustração da formação <i>flocking</i>	48
Figura 14 – Como o particionamento se apresenta num espaço bidimensional	53
Figura 15 – Como a acurácia de classificação depende do valor de α_1 e do número de iterações t	55
Figura 16 – Mapas de classificação em duas bases de dados artificiais: “duas gaussianas” e “arcos entrelaçados”	57
Figura 17 – Ilustração do processo de segmentação	65
Figura 18 – Imagens empregadas nos experimentos	66
Figura 19 – Exemplos de resultados obtidos com a técnica baseada na dinâmica coletiva de superpixels	68
Figura 20 – Efeitos da obtenção de diferentes quantidades de superpixels	69
Figura 21 – Visão geral do efeito causado por diferentes ajustes de parâmetros: quantidade de partículas (superpixels) vs. valor do expoente vs. k	70
Figura 22 – Análise de parâmetros para a imagem da estrela	71
Figura 23 – Análise de parâmetros para a imagem do lago	72

Figura 24 – Análise de parâmetros para a imagem da torre, particionada em dois segmentos	73
Figura 25 – Análise de parâmetros para a imagem da torre, particionada em quatro segmentos	74
Figura 26 – Além de isolar o céu dos elementos de frente, este exemplo mostra um bom refinamento e um refinamento errado que quebra o céu em dois segmentos	76
Figura 27 – Exemplo de como uma região de cor heterogênea está sujeita à fragmentação	77
Figura 28 – Diferentes interpretações das construções	78
Figura 29 – Duas formas distintas contra um plano de fundo amarelo	79
Figura 30 – Exemplo de colisão entre diferentes partes da imagem	79
Figura 31 – Exemplo de evitar colisão com o uso de mais dimensões	80
Figura 32 – Comparação de acurácia com a imagem da estrela	82
Figura 33 – Comparação de acurácia com a imagem do lago	83
Figura 34 – Comparação de acurácia com a imagem da torre, considerando dois segmentos	84
Figura 35 – Comparação de acurácia com a imagem da torre, considerando quatro segmentos	84
Figura 36 – Ilustração do processo de alinhamento	90
Figura 37 – Evolução da entropia da rede apresentada na Figura 36: 4 comunidades, cada uma formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$	93
Figura 38 – Visão geral dos resultados de detecção de comunidades	94
Figura 39 – Visão geral dos resultados de detecção de comunidades numa situação um pouco mais fácil: $p_{in} = 0,66$	95
Figura 40 – Valores ARI e de modularidade ao longo dos diferentes ciclos (<i>rounds</i>)	98
Figura 41 – Comparação entre remover 10 arestas ($\approx 0,25\%$) e 40 arestas ($\approx 1\%$) por ciclo numa população de 50 redes diferentes	99
Figura 42 – Valores ARI e de modularidade ao longo de diferentes ciclos, utilizando-se redes de diferentes probabilidades de conexão intra-comunidade p_{in} : 0,62, 0,64, 0,66 e 0,68	100
Figura 43 – Resultados obtidos para um amplo conjunto de diferentes redes	101
Figura 44 – Boxplots destacando os diferentes resultados para certas probabilidades p_{in}	102
Figura 45 – Evolução da entropia numa rede 10 vezes maior que aquelas estudadas na seção anterior: 4 comunidades, cada uma formada por 2000 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$	104
Figura 46 – Resultados obtidos numa rede 10 vezes maior: 4 comunidades, cada uma formada por 2000 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$	104
Figura 47 – Resultados obtidos sobre uma população de 50 redes distintas	105
Figura 48 – Evolução da entropia sobre a rede mostrada na Figura 36: 4 comunidades, cada uma formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$	107

Figura 49 – Resultados obtidos rodando-se cada simulação até $t = 800$	108
Figura 50 – Resultados com uma rede desbalanceada formada por quatro comunidades, cada uma contendo 800, 400, 200 e 100 vértices, $\langle k \rangle \approx 10$ e $p_{\text{in}} = 0,65$. . .	108
Figura 51 – Resultados obtidos com redes desbalanceadas formadas por quatro comunidades, cada uma contendo 800, 400, 200 e 100 vértices, $\langle k \rangle \approx 10$ e $p_{\text{in}} = 0,65$	109
Figura 52 – Componentes conexas após certo número de ciclos de remoção de arestas . . .	110
Figura 53 – Ilustração do processo dinâmico modelado, em diferentes estágios	127
Figura 54 – Índice Rand em função do expoente	129
Figura 55 – Mapa de distâncias entre as partículas para diferentes valores de expoente e diferentes raios (tempo de evolução)	130
Figura 56 – Disposição inicial aleatória das partículas.	136
Figura 57 – Disposição final das partículas, sem o emprego do limiar d_0	137
Figura 58 – Disposição final das partículas, empregando-se o limiar $d_0 = 8$	138
Figura 59 – Dados utilizados nos experimentos aqui apresentados	139
Figura 60 – Resultados obtidos com a aplicação do algoritmo Cluster Fast Greedy . . .	141
Figura 61 – Resultados obtidos com a aplicação do algoritmo Louvain	142
Figura 62 – Representação em forma de cores das distâncias entre as partículas no momento em que o processo dinâmico é interrompido	143
Figura 63 – Resultados obtidos com a técnica proposta neste trabalho	144
Figura 64 – Resultados da aplicação da técnica proposta neste trabalho sobre uma rede com efeito “small world”	145
Figura 65 – Resultados obtidos com a aplicação de outros algoritmos da literatura . . .	146

LISTA DE TABELAS

Tabela 1 – Valores atribuídos por $\text{type}(\mathbf{x}_i, \mathbf{x}_j)$	51
Tabela 2 – As bases de <i>benchmark</i> utilizadas nos experimentos, como proposto por Chapelle, Schölkopf e Zien (2006).	58
Tabela 3 – Referências das outras técnicas usadas para comparação.	58
Tabela 4 – Erros de classificação nas bases de dados propostas por Chapelle, Schölkopf e Zien (2006)	59

LISTA DE ABREVIATURAS E SIGLAS

ARI	Índice Rand Ajustado (<i>Adjusted Rand Index</i>)
CFG	<i>Cluster Fast Greed</i>
EM	<i>Expectation-Maximization</i>
k-NN	<i>k-Nearest Neighbors</i>
PCA	<i>Principal Component Analysis</i>
PSO	Otimização por Nuvem de Partículas (<i>Particle Swarm Optimization</i>)
SVM	Máquina de Vetores de Suporte (<i>Support Vector Machine</i>)

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Motivação e Objetivos	22
1.2	Organização do Documento	24
2	CONCEITOS, TÉCNICAS E TRABALHOS RELEVANTES	27
2.1	Sistemas Dinâmicos Coletivos	28
2.2	Aprendizado Semissupervisionado	32
2.2.1	<i>Aprendizado Semissupervisionado via Regularização Local e Global</i>	35
2.2.2	<i>Aprendizado Semissupervisionado via Caminhada de Partículas em Rede</i>	39
2.2.3	<i>Aprendizado Semissupervisionado via Movimentação dos Dados pelo Espaço de Atributos</i>	40
3	APRENDIZADO SEMISSUPERVISIONADO VIA DINÂMICA COLETIVA DE PARTÍCULAS	47
3.1	Descrição do Modelo	48
3.2	Complexidade Computacional	51
3.3	Resultados Experimentais	56
3.3.1	<i>Bases de Dados Artificiais</i>	56
3.3.2	<i>Bases de Benchmark</i>	57
4	SEGMENTAÇÃO DE IMAGENS VIA DINÂMICA COLETIVA DE SUPERPIXELS	61
4.1	Descrição do Modelo	62
4.2	Resultados Experimentais	66
4.2.1	<i>Visão Geral dos Resultados</i>	66
4.2.2	<i>Análise de Parâmetros</i>	69
4.2.3	<i>Resultados Comparativos</i>	81
5	DETECÇÃO DE COMUNIDADES VIA REMOÇÃO ITERATIVA DE ARESTAS	85
5.1	Descrição do Modelo	88
5.1.1	<i>O Modelo Dinâmico de Alinhamento de Partículas</i>	88
5.1.2	<i>O Processo Iterativo de Remoção de Arestas</i>	89

5.2	Resultados Experimentais	91
5.2.1	<i>Evolução da Entropia ao longo do Tempo</i>	92
5.2.2	<i>Visão Geral dos Resultados de Detecção de Comunidades</i>	92
5.2.3	<i>Análise de Parâmetros</i>	96
5.2.4	<i>Complexidade Computacional</i>	103
5.2.5	<i>Critério de Parada do Sistema Dinâmico</i>	106
5.2.6	<i>Experimentos com Comunidades Desbalanceadas</i>	106
6	CONCLUSÕES E TRABALHOS FUTUROS	111
REFERÊNCIAS		115
APÊNDICE A	ADAPTAÇÃO AO CENÁRIO NÃO-SUPERVISIONADO	125
A.1	Descrição do Modelo	125
A.2	Resultados Experimentais	126
A.3	Discussão dos Resultados	131
APÊNDICE B	REDES COM TOPOLOGIA ESPACIAL E AGRUPAMENTO DE DADOS	133
B.1	Descrição do Modelo	133
B.2	Experimentos: Quatro Comunidades com Diferentes Densidades . .	135
B.3	O Efeito “Small World”	140

INTRODUÇÃO

Tecnologias de aquisição de dados têm avançado muito nos últimos anos, disponibilizando quantidades massivas de dados nas diversas áreas da ciência e da indústria. Tecnologias capazes de processar todo esse volume de dados, porém, não tem acompanhado no mesmo ritmo. Logo, torna-se imprescindível o desenvolvimento e aprimoramento de ferramentas automatizadas e inteligentes capazes de processar esse alto volume de dados.

Resumidamente, o *aprendizado de máquina* é o campo responsável pelo desenvolvimento dos métodos que tornam os computadores capazes de aperfeiçoar seu desempenho, em determinada tarefa, por meio de exemplos ou experiências anteriores (MITCHELL, 1997; ALPAYDIN, 2004; BISHOP, 2006; FACELI *et al.*, 2011). Em outras palavras, como o próprio nome sugere, é o campo responsável por tornar os computadores capazes de “aprender” com os dados adquiridos. Vastas são suas aplicações, dentre as quais: mineração de dados, reconhecimento de faces e fala, e robótica (MITCHELL, 1997; ALPAYDIN, 2004; BISHOP, 2006; FACELI *et al.*, 2011).

Os métodos de aprendizado podem ser divididos em *supervisionado* e *não-supervisionado* (MITCHELL, 1997; HASTIE; TIBSHIRANI; FRIEDMAN, 2001; ALPAYDIN, 2004; BISHOP, 2006; FACELI *et al.*, 2011). Tarefas de *classificação* enquadram-se no aprendizado supervisionado (conhecido também como aprendizado preditivo). Nelas, trabalha-se com coleções de *dados* (também chamados *objetos*, *observações*, *instâncias* ou *itens*) previamente classificados. O objetivo é encontrar uma função a ser posteriormente empregada para classificar novos dados, com base nos antigos já classificados. A essa função é dado o nome de *classificador*, e representa uma superfície de decisão que separa as diferentes classes. O aprendizado não-supervisionado (ou aprendizado descritivo), por sua vez, parte de coleções de objetos não-rotulados, onde classes não estão previamente estabelecidas. O objetivo é descobrir padrões que regem a distribuição desses objetos. O *agrupamento de dados* (*clustering*, ou simplesmente *agrupamento*) figura como uma das tarefas do aprendizado não-supervisionado e, como o nome sugere, procura agrupar os objetos de acordo com sua similaridade (JAIN; MURTY; FLYNN, 1999; XU; Wunsch

II, 2005; FACELI *et al.*, 2011).

O aprendizado pode ainda ser denominado *semisupervisionado* (SEEGER, 2002; CHAPPELLE; SCHÖLKOPF; ZIEN, 2006; ZHU, 2008; ZHU; GOLDBERG, 2009; WANG, 2010; SILVA; ZHAO, 2012b). Aqui, parte-se de uma mistura de alguns dados rotulados (classificados) em meio a muitos não-rotulados. Muitos dos problemas reais caem nessa configuração, pois a rotulação de dados é uma tarefa onerosa e tipicamente requer a intervenção de profissionais especialistas. Objetiva-se classificar os dados não-rotulados em função dos rotulados e do próprio padrão de distribuição dos não-rotulados. Essa tarefa é discutida em detalhes na Seção 2.2.

Sistemas dinâmicos coletivos são caracterizados por um volumoso conjunto de indivíduos, cada qual um sistema dinâmico por si só, de modo que todos eles agem coletivamente, ou seja, a ação de cada indivíduo é influenciada pela ação dos vizinhos. Uma característica notável desses sistemas é que, sob determinadas circunstâncias, um padrão organizado e global pode “emergir” espontaneamente das interações locais entre os indivíduos, fenômeno conhecido como *emergência* (VICSEK *et al.*, 1995; TONER; TU, 1995; TONER; TU, 1998; BONABEAU; DORIGO; THERAULAZ, 1999; KENNEDY; EBERHART, 2001; ENGELBRECHT, 2005; CHATÉ *et al.*, 2008; GINELLI *et al.*, 2010; MENON, 2010; RAMASWAMY, 2010; MISHRA *et al.*, 2012; NGO; GINELLI; CHATÉ, 2012; PESHKOV *et al.*, 2012; SUMINO *et al.*, 2012; TONER, 2012a; TONER, 2012b; VICSEK; ZAFEIRIS, 2012; NAGAI *et al.*, 2015). Tais sistemas são auto-organizáveis, portanto não há uma autoridade central (um “líder”) planejando e ditando como o sistema deve se comportar.

Um dos grandes desafios dos sistemas dinâmicos coletivos é que não se pode “programar” o padrão global emergente, mas apenas as interações locais entre os indivíduos. Em outras palavras, é difícil compreender as leis locais que originam um fenômeno coletivo global, se somente o próprio fenômeno global pode ser observado. Nesse sentido, cientistas têm procurado por características comuns, talvez universais, presentes no movimento coletivo de animais, bactérias, células em geral, motores moleculares etc., caracterizando o que vem sendo chamado de *matéria ativa* (VICSEK *et al.*, 1995; TONER; TU, 1995; TONER; TU, 1998; CHATÉ *et al.*, 2008; GINELLI *et al.*, 2010; MENON, 2010; RAMASWAMY, 2010; MISHRA *et al.*, 2012; NGO; GINELLI; CHATÉ, 2012; PESHKOV *et al.*, 2012; SUMINO *et al.*, 2012; TONER, 2012a; TONER, 2012b; VICSEK; ZAFEIRIS, 2012; NAGAI *et al.*, 2015). Trata-se de um campo de pesquisa interdisciplinar que vem ganhando muita atenção. Toda essa pesquisa nos equipa, hoje, com resultados práticos e teóricos, embora haja muito ainda a ser feito.

1.1 Motivação e Objetivos

Este trabalho toma a hipótese de que sistemas dinâmicos coletivos são um mecanismo poderoso na solução de uma ampla classe de problemas computacionais complexos, o que inclui as tarefas do aprendizado de máquina, cujas soluções ótimas não podem ser obtidas

exaustivamente (por força bruta), mas admitem soluções boas.

Nesse sentido, o presente trabalho de doutorado objetiva o desenvolvimento e a análise de modelos dinâmicos coletivos para tarefas do aprendizado de máquina, especificamente tarefas não-supervisionada e semissupervisionada. Nesses modelos, os dados tipicamente se movimentam pelo espaço de atributos, de modo que a força atuante sobre cada objeto seja regida pela *localização* e pela *velocidade* dos objetos vizinhos. A consideração da localização é tradicionalmente encontrada em outros trabalhos da literatura, que se baseiam em comportamento coletivo para a solução de problemas complexos (BONABEAU; DORIGO; THERAULAZ, 1999; DORIGO; STÜTZLE, 2004; ENGELBRECHT, 2005). Nesses trabalhos, entretanto, é comum fazer os objetos convergirem para simples pontos no espaço, o que acaba ignorando a essência da dinâmica coletiva: a formação de padrões. Além disso, esse tipo de modelagem torna o sistema muito rígido e pouco robusto. Já a consideração da velocidade, por outro lado, é ainda pouco explorada na literatura de aprendizado de máquina, embora tal característica já venha sendo estudada nas outras ciências (MISHRA *et al.*, 2012; VICSEK; ZAFEIRIS, 2012; NAGAI *et al.*, 2015). Esses modelos estabelecem claramente as leis que regem a velocidade dos objetos, permitindo assim a formação de padrões diversos. Por outro lado, o objetivo desses trabalhos é compreender a dinâmica coletiva em si, e não necessariamente fazer com que os padrões formados correspondam a soluções de problemas. Assim, neste trabalho de doutorado, objetiva-se combinar o modelo de localização com o modelo de velocidade, aproveitando-se as vantagens de cada um. Espera-se que o sistema dinâmico coletivo assim modelado seja conduzido a um estado cujo padrão formado por seus indivíduos realce padrões subjacentes do conjunto de dados. Além disso, o próprio caráter dinâmico e auto-organizável dos modelos tende a torná-los robustos e fazer com que as informações geradas durante o processo (valores das variáveis do sistema) sejam ricas e possam, por exemplo, revelar características para realizar *soft labeling* e determinar classes sobrepostas.

A intenção de resolver problemas de aprendizado de máquina recorrendo-se à literatura interdisciplinar de dinâmica coletiva é, mais que uma inspiração, um esforço em aproveitar o conhecimento que vem sendo produzido nesse campo de intensa pesquisa. Mais à frente, na Seção 2.1, mostra-se como os estudos sobre sistemas dinâmicos coletivos são conduzidos em outras ciências. Uma característica fundamental desses sistemas é que as partículas comportam-se conforme o comportamento em sua vizinhança. Como exemplo, é mostrado o trabalho de Vicsek *et al.* (1995), que modela o movimento das partículas em função da direção tomada pelas suas vizinhas.

Passemos então a uma descrição mais formal da classe dos modelos que são desenvolvidos e analisados neste trabalho. Seja $\{\mathbf{x}_i\}$ um conjunto de dados, onde cada objeto $\mathbf{x}_i$ é um vetor no espaço de atributos, portanto representa a localidade do i -ésimo item de dados. Seja $\mathbf{v}_i(t)$ o vetor que representa a velocidade tomada pelo objeto $\mathbf{x}_i$ no instante t . E seja $\text{viz}(\mathbf{x}_i(t))$ uma certa vizinhança de $\mathbf{x}_i$ também no instante t . Objetiva-se, então, modelar e analisar sistemas dinâmicos

com a seguinte forma:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \alpha_1 \cdot \mathbf{v}_i(t) \quad (1.1)$$

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \alpha_2 \cdot f\left(\left\{\mathbf{x}_j(t), \mathbf{v}_j(t) \mid \mathbf{x}_j(t) \in \text{viz}(\mathbf{x}_i(t))\right\}\right) \quad (1.2)$$

Os coeficientes α_1 e α_2 definem o quão suave é o processo. $f(\cdot)$ representa uma função genérica que leva em conta tanto a localização quanto a velocidade dos objetos vizinhos. A modelagem de $f(\cdot)$ é justamente um dos objetos de investigação deste trabalho. Objetiva-se explorar a influência de diferentes funções $f(\cdot)$ no comportamento global do sistema, e como isso reflete os padrões originalmente presentes no conjunto de dados. Por fim, ajustes no modelo são feitos para discriminar objetos rotulados dos não-rotulados, pois também trata-se de um modelo a ser empregado em tarefas do aprendizado semissupervisionado.

A composição das Equações 1.1 e 1.2 resulta num sistema dinâmico não-linear. Sabendo-se que essa não-linearidade introduz um comportamento complexo ao sistema, o exame de suas variáveis ao longo do tempo pode revelar informações valiosas sobre o conjunto de dados.

Além do mais, esse modelo descreve o comportamento de uma ampla classe de sistemas formados por partículas autopropulsionadas. Portanto, ele apresenta generalidade em sistemas dinâmicos coletivos, e seu estudo pode contribuir não só para o aprendizado de máquina — principal objetivo deste trabalho —, como também para o próprio estudo no campo da dinâmica coletiva.

Outra característica notável dos sistemas que são aqui estudados é que eles são intrinsecamente flexíveis. Quando empregados em tarefas do aprendizado de máquina, por exemplo no caso do aprendizado semissupervisionado, eles não conduzem a uma rotulação rígida dos dados, mas permitem aos objetos diferentes graus de pertinência a diferentes classes, característica conhecida como *soft labeling* (YANG, 1993; HÜLLERMEIER, 2005; El-Zahhar, 2010).

Importante dizer que, originalmente, este trabalho de doutorado tinha como objetivo a concepção e análise de modelos dinâmicos coletivos especificamente para tarefas do aprendizado semissupervisionado. Tais modelos apresentam algumas complicações intrínsecas decorrentes do papel diferenciado exercido pelos itens rotulados. Assim, o desenvolvimento da pesquisa já previa também a análise de versões mais simples e fundamentais do modelo. Essas versões mais fundamentais caracterizam naturalmente modelos aplicáveis em tarefas do aprendizado não-supervisionado, tarefas essas que acabaram sendo bastante trabalhadas no decorrer deste projeto de doutorado.

1.2 Organização do Documento

Este documento está organizado da seguinte forma. No Capítulo 2, as duas principais áreas envolvidas na concepção original deste projeto (aprendizado semissupervisionado e sis-

temas dinâmicos coletivos) são discutidas em maiores detalhes. Trabalhos relacionados são apresentados nesse mesmo capítulo. A partir de então, os capítulos subsequentes são dedicados a apresentar os resultados obtidos ao longo desta pesquisa. Enquanto os Capítulos 3, 4 e 5 trazem uma discussão detalhada, os Apêndices A e B apresentam resultados relativamente preliminares, por isso foram deixados no final do documento, como material complementar. Com exceção dos Apêndices A e B, a ordem dos resultados reflete a sequência em que foram desenvolvidos, que de forma geral, parte de um trabalho dedicado ao aprendizado semissupervisionado, cujo modelo é relativamente complexo, e vai conduzindo a trabalhos cujos modelos são mais simples e fundamentais. Nesse sentido, a técnica dinâmica coletiva dedicada ao aprendizado semissupervisionado é apresentada no Capítulo 3. Um estudo em que retira-se o papel diferenciado dos itens rotulados e conduz-se ao caso não-supervisionado é apresentado no Apêndice A. Uma aplicação em segmentação de imagens digitais é apresentada no Capítulo 4. Por fim, no Capítulo 5 e no Apêndice B, apresentam-se técnicas dinâmicas coletivas com aplicação em detecção de comunidades em redes. Conclusões e trabalhos futuros são apresentados no Capítulo 6.

CONCEITOS, TÉCNICAS E TRABALHOS RELEVANTES

Neste capítulo, as duas principais áreas envolvidas na concepção original deste projeto (aprendizado semissupervisionado e sistemas dinâmicos coletivos) são discutidas com um pouco mais de detalhes. Como foi mencionado no capítulo anterior, o desenvolvimento desta pesquisa previa a análise de versões mais fundamentais do modelo dinâmico inicialmente concebido para o aprendizado semissupervisionado, e portanto o próprio desenvolvimento da pesquisa faria com que outros tópicos também fossem abordados, tópicos esses que podem ser caracterizados como aprendizado não-supervisionado, e que inclui tarefas de segmentação de imagens digitais e detecção de comunidades em redes. Vale notar também que uma revisão detalhada de agrupamento de dados, que é uma típica tarefa não-supervisionada, faz parte da dissertação de mestrado do mesmo autor desta tese de doutorado:

Roberto Alves Gueleri. **Agrupamento de dados baseado em comportamento coletivo e auto-organização**. Orientador: Zhao Liang. Dissertação de mestrado desenvolvida no Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. Ano: 2013.

Os conceitos e trabalhos apresentados neste capítulo procuram trazer uma visão geral que permita entender a proposta deste projeto de doutorado. Conceitos mais específicos, porém, são apresentados e discutidos diretamente onde são aplicados ao longo dos capítulos de resultados.

Na Seção 2.1, apresenta-se como os estudos sobre dinâmica coletiva são conduzidos nas outras ciências. Nessa mesma seção, apresenta-se uma visão geral de trabalhos que procuram tratar problemas do aprendizado de máquina por meio da dinâmica coletiva. Na Seção 2.2, apresentam-se quais são as tarefas envolvidas no aprendizado semissupervisionado. Por fim, nas Seções 2.2.1, 2.2.2 e 2.2.3, três técnicas de aprendizado semissupervisionado são descritas

em mais detalhes, sendo as duas últimas mais relacionadas com este projeto de doutorado, empregando a ideia de dinâmica coletiva. Em especial, a técnica da Seção 2.2.3 teve participação do autor desta tese e constitui um trabalho que serviu de base para este trabalho de doutorado.

2.1 Sistemas Dinâmicos Coletivos

Como já mencionado, cientistas têm procurado por características comuns, talvez universais, presentes no movimento coletivo observado em sistemas naturais (matéria ativa). Sumino *et al.* (2012) evidenciam a existência de certas classes universais do fenômeno de movimento coletivo (*flocking*). Eles realizaram experimentos com microtúbulos (motores moleculares) capazes de se alinhar quando colidem entre si. Em altas concentrações, um fenômeno de auto-organização se torna visível numa macroescala: a população de microtúbulos dá origem a um retículo de vórtices (turbilhões ou redemoinhos). Os autores confirmaram os mesmos resultados modelando matematicamente e simulando o comportamento dos microtúbulos, o que permitiu conhecer os eventos locais que originam a organização global observada. Enquanto trabalhos como o de Sumino *et al.* (2012) trazem uma visão mais empírica, outros como os de Toner e Tu (1995), Toner e Tu (1998), Ramaswamy (2010), Peshkov *et al.* (2012) e Toner (2012a), Toner (2012b) enfatizam também um lado mais teórico do assunto.

A fim de ilustrar tais estudos sobre movimento coletivo, consideremos o trabalho de Vicsek *et al.* (1995). Trata-se de um trabalho bastante referenciado sobre o assunto (TONER; TU, 1995; CHATÉ *et al.*, 2008; GINELLI *et al.*, 2010; MENON, 2010; RAMASWAMY, 2010; NGO; GINELLI; CHATÉ, 2012; PESHKOV *et al.*, 2012; SUMINO *et al.*, 2012). Nele, os autores propõem um modelo dinâmico e coletivo formado por partículas autopropulsionadas (semelhante aos motores moleculares). Seu intuito é estudar as diferentes fases que o sistema exhibe. Mostra-se que a partir de um modelo relativamente simples, com interações simples entre os indivíduos, o sistema é capaz de exibir um comportamento global mais complexo, resultando no movimento coordenado das partículas. Em algumas circunstâncias, o sistema parte de um estado *sem transporte*¹ (média das velocidades aproximadamente zero) e espontaneamente entra num estado de transporte.

Cada partícula mantém velocidade absoluta constante e a cada iteração toma a direção média das partículas vizinhas. Simulações foram realizadas em uma região quadrada de lado L , considerando uma *condição de contorno periódica* (*periodic boundary condition*)². Inicialmente (tempo $t = 0$), N partículas são aleatoriamente dispostas, todas com a mesma velocidade absoluta v mas com direções θ aleatórias. A posição da i -ésima partícula é atualizada conforme

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t \quad (2.1)$$

¹ Conceito físico de *fenômenos de transporte*.

² Uma *condição de contorno periódica* é utilizada para simular um sistema homogêneo e infinitamente grande pela simulação de apenas uma pequena região (uma célula). Os objetos que saem da célula entram imediatamente na mesma célula através da face oposta, mantendo a mesma velocidade.

Figura 1 – Velocidade das partículas após algum tempo de evolução, para diferentes valores de densidade e ruído. As três simulações empregam $N = 300$ e $v = 0,03$. (a): Densidade e ruído altos ($L = 7$, $\eta = 2,0$), as partículas se movimentam aleatoriamente com alguma correlação. (b): Densidade e ruído baixos ($L = 25$, $\eta = 0,1$), as partículas tendem a formar grupos. (c): Densidade alta e ruído baixo ($L = 5$, $\eta = 0,1$), o movimento se torna completamente ordenado, todas as partículas tomam espontaneamente uma mesma direção (estado de transporte). Adaptada de [Vicsek et al. \(1995\)](#).

A nova velocidade de cada partícula, $\mathbf{v}_i(t+1)$, continua tendo valor absoluto v , mas direção $\theta(t+1)$ conforme

$$\theta(t+1) = \langle \theta(t) \rangle_r + \Delta\theta \quad (2.2)$$

onde $\langle \theta(t) \rangle_r$ denota a direção média das partículas (incluindo a partícula i) dentro de um círculo de raio r centrado na partícula i . $\Delta\theta$ denota uma perturbação aleatória cujo valor se encontra no intervalo $[-\eta/2, \eta/2]$. Ou seja, $\Delta\theta$ representa ruído. Fixando $\Delta t = 1$ e $r = 1$, o modelo apresenta três parâmetros: a velocidade v , o nível de ruído η e a densidade inicial de partículas ρ , tal que $\rho = N/L^2$. Na Figura 1, exibe-se o resultado de algumas simulações.

Diz-se que há uma *transição de fase* quando a variação de algum parâmetro resulta em diferentes estados qualitativos (*fases*). No caso aqui mostrado, pode-se dizer que o sistema apresenta duas fases: fase sem transporte e fase de transporte. Para mensurar e evidenciar tal característica, adota-se o *parâmetro de ordem* v_a , a média normalizada das velocidades das partículas (valor absoluto):

$$v_a = \frac{1}{Nv} \left\| \sum_{i=1}^N \mathbf{v}_i \right\| \quad (2.3)$$

$v_a \approx 0$ significa que não há transporte, enquanto $v_a \approx 1$ significa que as partículas tomaram uma mesma direção, portanto há transporte. Na Figura 2, mostra-se v_a em função do nível de ruído η . O gráfico exibe justamente o intervalo de valores η para o qual há a transição entre as duas fases do sistema. Observa-se que a diminuição gradual do nível de ruído η leva à transição do movimento desordenado ao movimento coerente das partículas. Nota: cada entrada (amostra) do gráfico corresponde a uma simulação independente das demais.

Outro trabalho bastante conhecido na literatura é o de [Reynolds \(1987\)](#), [Reynolds \(1999\)](#). Nele, propõe-se o chamado modelo de *boids*, que procura modelar agentes autônomos, cujo

Figura 2 – Valor absoluto da média normalizada das velocidades (v_a) em função do nível de ruído η , para diferentes quantidades de partículas N . A densidade ρ é mantida constante. Portanto, para $N = 40$, $L = 3$, 1, para $N = 100$, $L = 5$... Adaptada de [Vicsek et al. \(1995\)](#).

Figura 3 – As três leis de interação do modelo de *boids*. Adaptada de [Reynolds \(1999\)](#).

objetivo primário é sua utilização em animação e games. Dentre as principais características desse modelo, três leis de interação são definidas a fim de garantir o movimento coordenado de grandes grupos desses agentes:

1. Alinhamento: cada agente tende a tomar a direção de deslocamento dos agentes vizinhos;
2. Aproximação (coesão): os agentes tendem a se aproximar uns dos outros; e
3. Separação: os agentes tendem a manter uma distância mínima uns dos outros, evitando sua colisão.

Essas três leis de interação são ilustradas na Figura 3.

No campo do aprendizado de máquina, vem sendo crescente o interesse em abordagens alternativas aos métodos tradicionais. Abordagens inspiradas na natureza, em especial algumas baseadas em grupos de indivíduos que interagem entre si (sistemas dinâmicos coletivos), vêm ganhando espaço e mostrando-se eficientes ([BLATT; WILSEMAN; DOMANY, 1997](#); [OMRAN; SALMAN; ENGELBRECHT, 2002](#); [ZHAO; Damiance Jr.; CARVALHO, 2005](#); [ABRAHAM;](#)

DAS; ROY, 2007; OLIVEIRA, 2008; GUELERI; ZHAO, 2011; GUELERI; ZHAO, 2012; SILVA; ZHAO, 2012c; SILVA; ZHAO, 2012b; SILVA; ZHAO, 2012a; SILVA; ZHAO; CUPERTINO, 2012; CUPERTINO; GUELERI; ZHAO, 2014).

Blatt, Wilseman e Domany (1997) desenvolveram um algoritmo de agrupamento de dados baseado nas propriedades físicas de um ferromagneto heterogêneo. O algoritmo, também denominado *super-paramagnetic clustering*, é fundamentado em conceitos da mecânica estatística. De maneira geral, um *spin* é atribuído a cada objeto, de modo que cada um desses *spins* será influenciado pelos *spins* dos objetos vizinhos. A influência entre cada par de objetos será tão maior quanto menor for a distância entre eles. O algoritmo faz uso do conceito de temperatura, tal que o sistema, a temperaturas muito baixas, encontra-se completamente ordenado, ou seja, todos os *spins* estão alinhados, enquanto a temperaturas muito altas, completamente desordenado. Em regimes intermediários, diferentes *clusters* são formados, cada um representando uma região que se torna ordenada. Dada uma temperatura, o agrupamento dos objetos levará em conta a correlação entre seus *spins*, de modo que um *cluster* será formado por um conjunto de objetos cujos *spins* se tornaram alinhados.

Cui, Gao e Potok (2006) propuseram uma técnica de agrupamento de documentos baseada na formação de *flocks*. Esse trabalho também leva em conta as três leis de interação do modelo de *boids* proposto por Reynolds (1987), Reynolds (1999): alinhamento, aproximação e separação. Cada documento (um vetor de atributos) desempenha o papel de um *boid* e é projetado num espaço bidimensional, com o intuito de reduzir a dimensionalidade e facilitar a visualização dos dados e resultados. Apesar dessa semelhança com a proposta deste projeto de doutorado, o trabalho de Cui apresenta diferenças significativas na modelagem.

Zhao, Damiance Jr. e Carvalho (2005) propuseram um método de agrupamento dinâmico e auto-organizável baseado na atração de objetos similares. Os objetos se movimentam pelo espaço de atributos, guiados por forças de atração exercidas entre eles. Assim, diferentes grupos de objetos convergirão para diferentes pontos, que são considerados os centros de cada *cluster*.

A Otimização por Nuvem de Partículas (*Particle Swarm Optimization*, ou PSO), técnica desenvolvida por Kennedy e Eberhart (1995), toma como inspiração o movimento organizado de grandes grupos de animais, especialmente pássaros e peixes (KENNEDY; EBERHART, 1995; KENNEDY; EBERHART, 2001; OMRAN; SALMAN; ENGELBRECHT, 2002; ENGELBRECHT, 2005; OLIVEIRA, 2008). Em linhas gerais, a PSO é constituída por uma população de partículas que explora o espaço de busca (espaço de soluções de um determinado problema) de maneira inicialmente aleatória. À medida que algumas partículas encontram regiões mais promissoras, são seguidas pelas demais. Entretanto, as partículas ainda mantêm certa liberdade para explorar novas regiões, procurando soluções ainda melhores.

Omran, Salman e Engelbrecht (2002) propuseram um algoritmo de agrupamento baseado na PSO. Cada partícula, representando cada solução do problema, é formada por um conjunto

de *centroides*³. Portanto, cada partícula representa uma partição dos dados. Assim, o algoritmo emprega a PSO na tentativa de melhorar cada solução obtida, gerando-se novas partições através da movimentação dos centroides. Importante ressaltar que, neste algoritmo, cada partícula não está vinculada nem a um objeto do conjunto de dados nem a um centroide, mas sim ao *conjunto* de todos os centroides de uma partição.

Outra técnica de aprendizado que vem ganhando bastante atenção é a de *competição de partículas em redes complexas* (SILVA; ZHAO, 2012c; SILVA; ZHAO, 2012b; SILVA; ZHAO, 2012a; SILVA; ZHAO; CUPERTINO, 2012). As *redes complexas* surgiram da representação de sistemas complexos em vários ramos da ciência, e são caracterizadas como redes (grafos) muito grandes e de topologia não-trivial (ALBERT; BARABÁSI, 2002; NEWMAN, 2003; BARRAT; BARTHÉLEMY; VESPIGNANI, 2008). A técnica de competição de partículas pode ser aplicada para detecção de comunidades⁴ ou para agrupamento, dependendo da natureza dos dados. Em essência, a tarefa de agrupamento pode ser vista como uma tarefa de detecção de comunidades, uma vez que qualquer conjunto de dados pode ser modelado como uma rede, baseando-se em alguma medida de similaridade. Pode ainda ser aplicada no aprendizado semissupervisionado, onde há alguns vértices já classificados (SILVA; ZHAO, 2012b) (ver Seção 2.2.2). A técnica consiste em dispor um conjunto de partículas sobre a rede. Cada partícula (ou grupo de partículas) carrega um rótulo. Iterativamente elas caminham pela rede, mesclando movimentos aleatórios e preferenciais, de modo a dominar vértices frequentemente visitados por elas. As partículas vão então disputando territórios (as comunidades), e ao final cada partícula (ou grupo de partículas) se estabiliza numa região, marcando-a com o respectivo rótulo.

2.2 Aprendizado Semissupervisionado

A tarefa do *aprendizado semissupervisionado* (especificamente o *aprendizado transdu-tivo*) pode ser entendida como: dado um conjunto contendo objetos rotulados (classificados) e não-rotulados (não-classificados), objetiva-se propagar os rótulos dos objetos previamente classificados aos ainda não-classificados (SEEGER, 2002; CHAPELLE; SCHÖLKOPF; ZIEN, 2006; ZHU, 2008; ZHU; GOLDBERG, 2009; WANG, 2010; SILVA; ZHAO, 2012b). Nesse processo, leva-se em conta não somente a localização dos dados rotulados, mas também o padrão de distribuição dos próprios dados não-rotulados.

A fração de dados inicialmente rotulados costuma ser bem pequena, e é aí que reside uma característica fundamental do aprendizado semissupervisionado. Tipicamente, a rotulação de dados envolve a intervenção de um profissional especialista, o que torna tal tarefa onerosa. Ao mesmo tempo, a existência de poucos dados previamente rotulados já constitui uma informação valiosa aos algoritmos de aprendizado semissupervisionado, dando a estes uma certa vantagem

³ Centroide é um vetor que representa a média de um conjunto de vetores (objetos).

⁴ A tarefa de detecção de comunidades em redes consiste em identificar grupos de vértices densamente conectados.

Figura 4 – Conjunto artificial de dados, cada qual caracterizado por dois atributos. Os dados estão divididos em três classes: duas gaussianas “alongadas” e sobrepostas ao centro, e um anel ao redor. Há um total de 1400 objetos, 2% deles previamente rotulados e destacados na figura. Seus símbolos denotam suas classes: triângulo e quadrado nas classes ao centro e pentágono no anel em volta.

em relação aos algoritmos de aprendizado não-supervisionado. Assim, o aprendizado semissupervisionado explora a vantagem de se ter dados rotulados por um especialista, mas mantendo viável seu custo, devido à necessidade de se rotular poucos dados.

Na Figura 4, mostra-se um típico conjunto de dados do aprendizado semissupervisionado. Nesse exemplo, mostra-se um conjunto artificial de dados, cada um caracterizado por dois atributos (conjunto bidimensional). O aprendizado semissupervisionado não se restringe, contudo, a conjuntos de dados semelhantes aos da figura. Ele abrange dados com atributos qualitativos (não-numéricos), dados caracterizados por redes, dados originados em fluxo (fluxo de dados) etc.

Como mencionado, a definição acima se refere ao *aprendizado transdutivo*, onde a tarefa é propagar os rótulos aos demais objetos já existentes. O aprendizado semissupervisionado não se restringe, porém, ao aprendizado transdutivo. Inclui também o *aprendizado indutivo*, onde objetiva-se encontrar uma função capaz de classificar novos objetos, ou seja, objetos indisponíveis durante o processo de aprendizado (CHAPELLE; SCHÖLKOPF; ZIEN, 2006).

Outro assunto normalmente tratado em aprendizado semissupervisionado refere-se aos chamados *pressupostos* (CHAPELLE; SCHÖLKOPF; ZIEN, 2006):

- *Pressuposto de suavidade (smoothness assumption)*: Se dois itens x_1 e x_2 estão perto um

do outro e localizados numa região de alta densidade, então seus rótulos y_1 e y_2 também devem ser semelhantes.

- *Pressuposto de agrupamento (cluster assumption)*: Se dois itens estão em um mesmo agrupamento (*cluster*), então espera-se que eles pertençam à mesma classe. Notar que o pressuposto de agrupamento é equivalente ao de suavidade caso um agrupamento seja definido como um conjunto de dados numa região densa. Com esses dois pressupostos, conclui-se que a fronteira de decisão que separa diferentes classes deve situar-se numa região de baixa densidade.
- *Pressuposto de “manifold” (manifold assumption)*: Um *manifold* é um espaço topológico que, localmente, assemelha-se ao espaço euclidiano. Globalmente, porém, pode ser completamente diferente do espaço euclidiano. Como exemplos, qualquer circunferência (somente o contorno) e qualquer esfera (somente a “casca”) são, respectivamente, *manifolds* unidimensional e bidimensional embutidos num espaço $\mathbb{R}^3$ (ou maior). Em aprendizado de máquina, pode ser que as classes tenham seus dados dispostos aproximadamente ao longo de *manifolds*.

Diversas abordagens têm sido propostas para aprendizado semissupervisionado. Os *modelos gerativos* procuram estimar os parâmetros das distribuições que supostamente “geraram” os padrões presentes no conjunto de dados. Há, por exemplo, extensões do algoritmo Expectation-Maximization (EM) para o aprendizado semissupervisionado (NIGAM; MCCALLUM; MITCHELL, 2006). O algoritmo, originalmente desenvolvido para o aprendizado não-supervisionado, assume que os dados foram gerados por sobreposição de distribuições gaussianas (como as duas classes interiores da Figura 4) (HASTIE; TIBSHIRANI; FRIEDMAN, 2001; XU; Wunsch II, 2009). Então o algoritmo EM procura determinar quais são os parâmetros (médias e variâncias) cujas gaussianas melhor refletem os dados observados. Qualquer informação adicional sobre os dados pode ser útil. Quando estendido ao aprendizado semissupervisionado, o algoritmo tira proveito justamente das informações dos rótulos.

Outros algoritmos procuram afastar, dos dados não-rotulados, as superfícies de decisão, consequentemente empurrando essas superfícies para regiões de baixa densidade. As Máquinas de Vetores de Suporte (*Support Vector Machines*, ou SVMs), empregadas no aprendizado supervisionado, procuram posicionar as curvas de decisão de modo a mantê-las o mais afastado possível dos objetos, neste caso, rotulados (HASTIE; TIBSHIRANI; FRIEDMAN, 2001; ALPAYDIN, 2004; FACELI *et al.*, 2011). Já as SVMs Transdutivas (*Transductive SVMs*, ou TSVMs) estendem essa ideia para o aprendizado semissupervisionado (JOACHIMS, 2006). Aqui, as superfícies de decisão não apenas separam os objetos (rotulados) de classes distintas, mas utilizam os não-rotulados (presentes em grande número) de modo a posicionar tais superfícies onde haja a menor concentração desses objetos (regiões de baixa densidade).

Há ainda os *métodos baseados em rede (grafo)*, que figuram entre os mais estudados em aprendizado semissupervisionado (BENGIO; DELALLEAU; ROUX, 2006; SINDHWANI; BELKIN; NIYOGI, 2006; ZHOU; SCHÖLKOPF, 2006; WANG, 2010; SILVA; ZHAO, 2012b). Cada objeto torna-se um vértice da rede. As arestas podem ou não ser ponderadas, e definem se há e com que intensidade há uma relação entre dois objetos. Há conjuntos de dados que são naturalmente descritos como uma rede, como a Internet, a Web, redes sociais etc. Mas mesmo conjuntos cujos dados são vetores de atributos dispostos num espaço n-dimensional podem ser convertidos para uma representação em rede, ligando-se cada objeto aos seus vizinhos mais próximos. Os métodos baseados em rede são usualmente transdutivos (BENGIO; DELALLEAU; ROUX, 2006; SINDHWANI; BELKIN; NIYOGI, 2006; ZHOU; SCHÖLKOPF, 2006). Eles estimam os rótulos dos dados já presentes, não generalizam funções para classificar novos dados, embora abordagens para indução de funções têm também sido propostas (SINDHWANI; BELKIN; NIYOGI, 2006; WANG, 2010). A técnica de competição de partículas em redes complexas, trabalhada por Silva e Zhao (2012b) e detalhada na Seção 2.2.2, também se enquadra como uma técnica de aprendizado semissupervisionado baseada em rede.

O aprendizado semissupervisionado encontra suas aplicações em cenários onde a obtenção dos dados é fácil, mas sua rotulação é onerosa e envolve, tipicamente, a intervenção de um especialista. Como exemplos (CHAPELLE; SCHÖLKOPF; ZIEN, 2006):

- Em reconhecimento de fala, é fácil gravar grandes quantidades de fala, mas sua rotulação exige que uma pessoa ouça cada unidade e anote sua transcrição.
- Bilhões de páginas Web estão disponíveis para processamento automático, mas classificá-las adequadamente exige que pessoas as leiam.
- Sequências de proteínas são hoje adquiridas em escala industrial, mas resolver sua estrutura tridimensional ou determinar as funções de uma simples proteína pode exigir anos de trabalho científico.

Nas seções seguintes, descrevem-se três técnicas de aprendizado semissupervisionado. A primeira, proposta por Wang (2010), pode ser entendida como uma técnica de aprendizado baseada em rede. Já as duas últimas são mais relacionadas com este projeto de doutorado pois empregam, direta ou indiretamente, a ideia de dinâmica coletiva, e foram desenvolvidas pelo grupo de pesquisa do qual o aluno faz parte.

2.2.1 Aprendizado Semissupervisionado via Regularização Local e Global

Nesta seção, procura-se apresentar as principais ideias de um *framework* proposto por Wang (2010), que leva em conta características globais e locais do padrão de distribuição dos

dados. Imaginando-se um cenário onde há somente itens rotulados (aprendizado supervisionado), um princípio comum é encontrar uma função f que minimize o seguinte *risco estrutural*:

$$\mathcal{J}_g = \sum_{i=1}^l \mathcal{L}(y_i, f(\mathbf{x}_i, \mathbf{w})) + \gamma \|f\|_{\mathcal{F}}^2 \quad (2.4)$$

onde cada $\mathbf{x}_i$ denota um item de dado (neste caso, rotulado, onde l é a quantidade de dados rotulados), $\mathbf{w}$ são parâmetros da função e y_i é o rótulo (já conhecido) de $\mathbf{x}_i$. $\mathcal{L}(\cdot, \cdot)$ é uma função de perda (ou custo) (*loss function*), que penaliza o quanto a resposta do classificador f se afasta do rótulo conhecido y_i . Já o termo $\gamma \|f\|_{\mathcal{F}}^2$, dito de modo informal e simplificado, representa uma penalização adicional que captura o quão suave é a função f pelo espaço de dados.

Adentrando agora o aprendizado semissupervisionado, ou seja, levando em conta os itens não-rotulados, uma extensão da Equação 2.4, como proposto por [Belkin, Niyogi e Sindhwani \(2006\)](#), seria:

$$\mathcal{R}_g = \sum_{i=1}^l \mathcal{L}(y_i, f(\mathbf{x}_i, \mathbf{w})) + \gamma_A \|f\|_{\mathcal{F}}^2 + \gamma \|f\|_{\mathcal{G}}^2 \quad (2.5)$$

onde o termo penalizador adicional $\gamma \|f\|_{\mathcal{G}}^2$ reflete a seguinte ideia (também de modo informal e simplificado): se dois itens $\mathbf{x}_1$ e $\mathbf{x}_2$ estão localizados próximos um ao outro, então espera-se que $f(\mathbf{x}_1)$ também não difira muito de $f(\mathbf{x}_2)$, ou seja, espera-se que eles recebam o mesmo rótulo. Novamente, $\gamma \|f\|_{\mathcal{G}}^2$ penaliza funções não-suaves.

Belkin aponta ainda que $\gamma \|f\|_{\mathcal{G}}^2$ pode ser aproximado por:

$$\mathcal{J} = \frac{\gamma}{n^2} \sum_{i,j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 w_{ij} = \frac{\gamma}{n^2} \mathbf{f}^T \mathbf{L} \mathbf{f} \quad (2.6)$$

onde n é a quantidade total de dados e w_{ij} são os pesos entre cada par de dados, especificamente, os pesos das arestas na respectiva *matriz de adjacência*. Tem-se também que $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^T$. Por fim, $\mathbf{L} = \mathbf{D} - \mathbf{W} \in \mathbb{R}^{n \times n}$ é a *matriz laplaciana*, tal que $\mathbf{W}$ é a matriz de pesos, $\mathbf{W}(i, j) = w_{ij}$, e $\mathbf{D}$ é uma matriz diagonal definida por $\mathbf{D}(i, i) = \sum_j w_{ij}$. Tipicamente, pesos altos são atribuídos a itens próximos um ao outro. Assim, penalizam-se os casos em que ambos w_{ij} e $|f(\mathbf{x}_i) - f(\mathbf{x}_j)|$ resultem em valores altos, pois, novamente, espera-se que um bom classificador não apresente transições abruptas, ou seja, mantenha um mesmo rótulo para itens muito similares.

Uma abordagem comum é definir um classificador f único e global para todo o espaço de dados. Essa é justamente a abordagem presente nas expressões até então aqui mostradas. Wang, entretanto, argumenta que tal escolha pode não ser ótima, pois pode não descrever adequadamente as diferenças existentes entre as diversas regiões do espaço de dados. Ao invés disso, propõe-se definir um classificador (local) para cada região, capaz então de capturar a estrutura presente naquela específica região. Para tanto, divide-se o espaço de dados em C regiões locais, e procura-se então minimizar a função de perda para cada região i :

$$\mathcal{J}_l^i = \sum_{j=1}^l K(\mathbf{x}_j - \mathbf{c}_i, \epsilon_i) \mathcal{L}(y_j, f(\mathbf{x}_j, \mathbf{w}_i)) + \gamma_A \|f_{\mathbf{w}_i}\|_{\mathcal{F}}^2 \quad (2.7)$$

Figura 5 – Dois exemplos de kernel locais. (a): Um kernel quadrado. (b): Um kernel suave. Ambos são centrados em c_0 e controlados pelo parâmetro ϵ_0 , que define sua largura. Obtida de Wang (2010).

onde $K(\mathbf{x}_j - \mathbf{c}_i, \epsilon_i)$ é um *kernel local* centrado em $\mathbf{c}_i$ e com largura ϵ_i . Aqui, $f_{\mathbf{w}_i}$ denota a função associada à região i , que recebe o parâmetro $\mathbf{w}_i$. Na Figura 5, são mostrados dois tipos de kernel comumente empregados. Como se pode ver, o kernel K indica a relevância de um item $\mathbf{x}_j$ para uma região local centrada em $\mathbf{c}_i$. Novamente, itens localizados perto do centro do kernel têm maior relevância e, conseqüentemente, provocam penalizações mais elevadas em $\mathcal{J}_l^i$ caso seu rótulo difira demais do rótulo associado à região i . Já os itens distantes praticamente anulam K , portanto não interferem no valor de $\mathcal{J}_l^i$. Combinando-se as C funções a serem minimizadas, tem-se então a função total:

$$\mathcal{J}_l = \sum_{i=1}^C \mathcal{J}_l^i \quad (2.8)$$

A Equação 2.7 ainda só leva em conta os l itens rotulados. A fim de comportar os itens não-rotulados, é necessário primeiramente introduzir o conceito de *pseudo-rótulo*, que pode ser entendido como uma versão não-rígida de um rótulo propriamente dito (um *soft label*). Assim, para todo item $\mathbf{x}_i$, é definido um pseudo-rótulo f_i . Então $\mathcal{J}_l$ pode ser redefinida como:

$$\mathcal{J}_l = \sum_{i=1}^l \mathcal{L}(f_i, y_i) + \lambda \cdot \left(\sum_{i=1}^C \sum_{j=1}^n K(\mathbf{x}_j - \mathbf{c}_i, \epsilon_i) \mathcal{L}(f_j, f(\mathbf{x}_j, \mathbf{w}_i)) + \gamma_A \|f_{\mathbf{w}_i}\|_{\mathcal{F}}^2 \right) \quad (2.9)$$

onde n é agora o tamanho de todo o conjunto de dados e $\lambda > 0$ é um parâmetro de balanço (*trade-off*). A minimização do primeiro termo faz f_i ficar suficientemente perto de y_i para os itens rotulados. Já a minimização do segundo termo tem as propriedades da Equação 2.7, mas agora levando em conta todos os dados, e considerando-se f_i como sendo o rótulo de $\mathbf{x}_i$.

Por fim, trazendo-se o termo $\gamma_l \|f\|_{\mathcal{F}}^2$ introduzido na Equação 2.5, que pode ser aproximado por $\frac{\gamma_l}{n^2} \mathbf{f}^T \mathbf{L} \mathbf{f} = \frac{\gamma_l}{n^2} \sum_{i,j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 w_{ij}$ (Equação 2.6), chega-se à expressão final a ser

minimizada:

$$\begin{aligned}\mathcal{R}_l &= \mathcal{J}_l + \gamma \|f\|_{\mathcal{F}}^2 = \\ &= \sum_{i=1}^l \mathcal{L}(f_i, y_i) + \lambda \cdot \left(\sum_{i=1}^C \sum_{j=1}^n K(\mathbf{x}_j - \mathbf{c}_i, \varepsilon_i) \mathcal{L}(f_j, f(\mathbf{x}_j, \mathbf{w}_i)) + \gamma_A \|f_{\mathbf{w}_i}\|_{\mathcal{F}}^2 \right) + \\ &\quad + \frac{\gamma}{n^2} \sum_{i,j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 w_{ij} \quad (2.10)\end{aligned}$$

Wang denomina o termo $\mathcal{J}_l$ como sendo a *regularização local* e o termo $\gamma \|f\|_{\mathcal{F}}^2$ como sendo a *regularização global*.

Por conseguinte, para se chegar num algoritmo de aprendizado semissupervisionado propriamente dito, três elementos devem ser definidos de maneira concreta:

1. Um kernel local K ;
2. Uma função de perda $\mathcal{L}(\cdot, \cdot)$; e
3. Um modelo para as funções f .

Wang, em seu trabalho, aplica cada um desses três elementos como descrito a seguir. O kernel local K é definido na k -vizinhança (mais próxima) de um ponto $\mathbf{x}_i$:

$$K(\mathbf{x}_j - \mathbf{c}_i, \varepsilon_i) = \begin{cases} 1 & \text{se } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \\ 0 & \text{caso contrário} \end{cases} \quad (2.11)$$

onde $\mathcal{N}(\mathbf{x}_i)$ denota a k -vizinhança de $\mathbf{x}_i$, $\mathbf{c}_i = 1/|\mathcal{N}(\mathbf{x}_i)| \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} \mathbf{x}_j$, $|\mathcal{N}(\mathbf{x}_i)|$ é a cardinalidade de $\mathcal{N}(\mathbf{x}_i)$, e $\varepsilon_i = 2 \max_{\mathbf{x}_j} \|\mathbf{x}_j - \mathbf{c}_i\|$. Tal kernel é do tipo quadrado mostrado na Figura 5. A função de perda $\mathcal{L}(\cdot, \cdot)$ é definida como a função de perda quadrática (*square loss function*) devido à sua simplicidade e ampla aplicação:

$$\mathcal{L}(y_i, f_i) = (y_i - f_i)^2 \quad (2.12)$$

$$\mathcal{L}(f_j, f(\mathbf{x}_j, \mathbf{w}_i)) = (f_j - f(\mathbf{x}_j, \mathbf{w}_i))^2 \quad (2.13)$$

E então seleciona-se o classificador linear como sendo o classificador local:

$$f(\mathbf{x}_j, \{\mathbf{w}_i, b_i\}) = \mathbf{w}_i^T \mathbf{x}_j + b_i \quad (2.14)$$

onde $\mathbf{w}_i$ e b_i são, respectivamente, o vetor de pesos e o viés (*bias*) do classificador linear.

Wang, após definir esses elementos, segue o trabalho com uma série de manipulações algébricas, a fim de chegar na expressão que define o vetor $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^T$, ou seja, o vetor que contém os rótulos de todos os itens, objetivo do classificador. Essencialmente, e de maneira bem geral, o trabalho consiste em tomar o vetor $\mathbf{f}$ que satisfaça $\partial \mathcal{R}_l / \partial \mathbf{f} = 0$, lembrando que $\mathcal{R}_l$ é uma função a ser minimizada. Maiores detalhes, claro, podem ser consultados no próprio trabalho de Wang (2010).

2.2.2 Aprendizado Semissupervisionado via Caminhada de Partículas em Rede

Como mencionado anteriormente, [Silva e Zhao \(2012b\)](#) propuseram uma técnica de aprendizado semissupervisionado baseada em competição de partículas em redes complexas. Em linhas gerais, um conjunto $\mathcal{K} = 1, \dots, K$ de partículas é inserido na rede. Cada partícula representa um rótulo a ser propagado. Pode haver mais de uma partícula carregando um mesmo rótulo, formando um “time” de cooperação. As partículas então caminham pela rede, mesclando movimentos aleatórios e preferenciais, na intenção de ampliar seu domínio sobre os vértices não-rotulados. Trata-se de uma competição por território. A cada passagem de uma partícula por um determinado vértice, essa partícula amplia a dominância do rótulo que carrega e ao mesmo tempo enfraquece a dominância de rótulos concorrentes. Ao final, o vértice não-rotulado receberá o rótulo que impôs maior dominância.

A matriz de transição associada ao movimento de uma partícula $k \in \mathcal{K}$ é definida como:

$$\mathbb{P}_{\text{transition}}^{(k)}(t) \triangleq (1 - S^{(k)}(t)) [\lambda \mathbb{P}_{\text{pref}}^{(k)}(t) + (1 - \lambda) \mathbb{P}_{\text{rand}}^{(k)}] + S^{(k)}(t) \mathbb{P}_{\text{rean}}^{(k)}(t) \quad (2.15)$$

Especificamente, $\mathbb{P}_{\text{transition}}^{(k)}(i, j, t)$ indica a probabilidade da partícula k realizar uma transição do vértice i ao j na iteração t . Notar que $\mathbb{P}_{\text{transition}}^{(k)}(t)$ é uma combinação de outras três matrizes de transição: movimento preferencial ($\mathbb{P}_{\text{pref}}^{(k)}(t)$), movimento aleatório ($\mathbb{P}_{\text{rand}}^{(k)}$) e movimento de reanimação ($\mathbb{P}_{\text{rean}}^{(k)}(t)$). Notar também que o movimento aleatório é independente da iteração t , pois tal matriz é determinada unicamente pela topologia da rede, não dependendo dos níveis de dominância já impostos até o momento. No movimento aleatório, a partícula tende a se mover para qualquer vértice adjacente, com igual probabilidade (distribuição uniforme). Já no movimento preferencial, a partícula dá prioridade a se mover para vértices sobre os quais a partícula já tenha maior nível de dominância. O parâmetro λ justamente define o balanceamento entre movimento aleatório e preferencial. A fim de impedir que uma partícula adentre um território “inimigo” e permaneça lá por muito tempo, um nível de energia é associado a cada partícula, sendo que essa energia diminui a cada vez que a partícula visita um vértice dominado por rótulo rival, mas também aumenta quando ela visita um vértice dominado por ela própria. Assim, a energia da partícula pode eventualmente se esgotar (a partícula se torna exausta) e então ela é “teleportada” para um de seus vértices, ou seja, vértices sobre os quais a partícula tem maior dominância. A matriz de reanimação define justamente esse movimento forçado de uma partícula exausta. A variável $S^{(k)}(t) \in \{0, 1\}$ simplesmente define o estado em que a partícula se encontra: 0 para ativa e 1 para exausta.

Dentre as análises realizadas, consideremos a análise dos parâmetros λ e Δ , sobre uma rede artificial com 1000 vértices, distribuídos em quatro classes de igual tamanho. Em média, cada vértice possui 40% de suas arestas conectando-o a vértices de classes distintas. 16 partículas são distribuídas na rede. Δ define o nível de incremento ou decremento de energia das partículas

Figura 6 – Acurácia de rotulação em função dos parâmetros λ e Δ , sobre uma rede artificial com 1000 vértices, distribuídos em quatro classes de igual tamanho. Em média, cada vértice possui 40% de suas arestas conectando-o a vértices de classes distintas. 16 partículas são distribuídas na rede. (a): Análise de λ fixando-se $\Delta = 0.07$. (b): Análise de Δ fixando-se $\lambda = 0.6$. Obtida de [Silva e Zhao \(2012b\)](#).

a cada iteração. Os resultados são apresentados na Figura 6.

A fim de ilustrar o processo de propagação de rótulos, considere a Figura 7. Apenas um vértice de cada classe é rotulado no início do processo, cada qual recebendo uma partícula. Já o resultado da classificação sobre outros dois conjuntos artificiais de dados é apresentado na Figura 8.

2.2.3 Aprendizado Semissupervisionado via Movimentação dos Dados pelo Espaço de Atributos

[Cupertino, Gueleri e Zhao \(2014\)](#) desenvolveram uma técnica de aprendizado semissupervisionado baseada na movimentação dos dados, que são impulsionados por forças de interação entre eles. Nessa técnica, os dados rotulados permanecem fixos, mas exercem forças de atração sobre os não-rotulados. Tão logo um objeto não-rotulado chegue suficientemente perto de um rotulado, o rótulo é propagado. Imediatamente, aquele originalmente não-rotulado — e que acabou de receber um rótulo — passa a agir como um rotulado, ou seja, permanece fixo e exerce forças de atração sobre os demais não-rotulados. O processo termina quando os rótulos são propagados a todos os dados. Como se pode notar, essa técnica é especificamente uma técnica de aprendizado transdutivo.

Seja $\{\mathbf{x}_i\}$ o conjunto de dados, onde cada objeto $\mathbf{x}_i$ é um vetor de atributos. Esse conjunto contém todos os objetos, rotulados e não-rotulados. Sejam também d_{ij} a distância euclidiana entre os objetos $\mathbf{x}_i$ e $\mathbf{x}_j$, e t o índice de iteração do sistema dinâmico (tempo). E denotaremos por $\mathcal{L}(t)$ e $\mathcal{U}(t)$ o conjunto dos objetos rotulados e não-rotulados no instante t , respectivamente.

Figura 7 – Ilustração do processo de classificação via competição de partículas. (a) $t = 0$, (b) $t = 100$, (c) $t = 200$ e (d) $t = 300$. Obtida de [Silva e Zhao \(2012b\)](#).

Figura 8 – Resultados de classificação sobre dois conjuntos artificiais de dados. (a) e (c): Configuração inicial, com alguns rótulos previamente definidos. (b) e (d): Classificação dos demais dados ao final do processo. Obtida de [Silva e Zhao \(2012b\)](#).

Figura 9 – O comportamento da função $f(d_{ij}) = d_{ij} \cdot e^{-\beta d_{ij}^2}$, para $\beta = 1$. Obtida de [Gueleri \(2013\)](#).

Então o movimento de cada objeto *não-rotulado* é regido por:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \alpha \cdot \sum_{\mathbf{x}_j \in \mathcal{L}(t)} d_{ij} \cdot e^{-\beta d_{ij}^2} \cdot \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{d_{ij}} \quad (2.16)$$

Nesta equação, $e^{-\beta d_{ij}^2}$ representa um campo gaussiano centralizado em $\mathbf{x}_i$ e com parâmetro β , $\beta > 0$. Multiplicando isso por d_{ij} , temos uma função como a apresentada na Figura 9. O termo $\frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{d_{ij}}$ representa um vetor unitário que define a direção da força sobre $\mathbf{x}_i$. Simplificando a Equação 2.16, temos:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \alpha \cdot \sum_{\mathbf{x}_j \in \mathcal{L}(t)} e^{-\beta d_{ij}^2} \cdot [\mathbf{x}_j(t) - \mathbf{x}_i(t)] \quad (2.17)$$

A somatória sobre $\mathbf{x}_j \in \mathcal{L}(t)$ denota a somatória sobre todos os objetos rotulados no instante t . Por fim, α define o quão suave é o processo. Podemos ainda escrever a versão contínua da Equação 2.17, fazendo $\alpha \rightarrow 0$:

$$\dot{\mathbf{x}}_i = \sum_{\mathbf{x}_j \in \mathcal{L}} e^{-\beta d_{ij}^2} \cdot [\mathbf{x}_j - \mathbf{x}_i] \quad (2.18)$$

Tão logo um objeto não-rotulado ($\mathbf{x}_i(t+1)$) atinja uma distância suficientemente próxima de um rotulado, o rótulo é propagado e ele passa a integrar o conjunto dos rotulados ($\mathcal{L}(t+1)$). Chamaremos essa distância de δ , tal que o objeto é rotulado quando $\|\mathbf{x}_i(t+1) - \mathbf{x}_j(t)\| \leq \delta$, para algum $\mathbf{x}_j(t) \in \mathcal{L}(t)$. Caso, após uma iteração, ele se torne suficientemente próximo a dois ou mais objetos rotulados, ele receberá o rótulo daquele mais próximo.

A fim de ilustrar como a técnica se comporta dinamicamente, considere o conjunto de dados mostrado na Figura 10. Algumas simulações foram realizadas sobre esse conjunto, para diferentes combinações dos parâmetros β e δ . Os resultados são mostrados nas Figuras 11 e 12.

Figura 10 – Conjunto artificial de dados, formado por duas gaussianas contendo 400 objetos cada, 20 deles previamente rotulados e destacados na figura: 11 na classe “triângulo” e 9 na classe “quadrado”. Cada objeto é descrito por dois atributos, resultando num conjunto bidimensional. Cada gaussiana tem variância 1 e seus centros distam 4 unidades um do outro. Obtida de [Gueleri \(2013\)](#).

Figura 11 – Comportamento dinâmico da técnica proposta, sobre o conjunto da Figura 10, empregando $\beta = 1$ e $p = 0,01$. A trajetória dos dados é mostrada. (a): $\delta = 0,1$, após algum tempo de evolução ($t = 200$). (b): $\delta = 0,1$, após a rotulação de todos os objetos ($t = 1659$). Taxa de erro: 2,9%. (c): $\delta = 0,2$, após a rotulação de todos os objetos ($t = 482$). Taxa de erro: 2,7%. Obtida de [Gueleri \(2013\)](#).

Figura 12 – Comportamento dinâmico da técnica proposta, sobre o conjunto da Figura 10, empregando $\beta = 50$ e $p = 0,01$. A trajetória dos dados é mostrada. (a): $\delta = 0,1$, após algum tempo de evolução ($t = 500$). (b): $\delta = 0,1$, após a rotulação de todos os objetos ($t = 6733$). Taxa de erro: 3,2%. (c): $\delta = 0,2$, após a rotulação de todos os objetos ($t = 2409$). Taxa de erro: 3,6%. Obtida de [Gueleri \(2013\)](#).

APRENDIZADO SEMISSUPERVISIONADO VIA DINÂMICA COLETIVA DE PARTÍCULAS

Os resultados decorrentes do presente trabalho de doutorado são apresentados a partir deste capítulo. O primeiro deles é uma técnica de aprendizado transdutivo, que em conformidade com a proposta desta tese de doutorado, é baseada na formação *flocking* (dinâmica coletiva) que pode ser observada na natureza. Precisamente, o modelo dinâmico proposto neste trabalho se assemelha com o modelo de *birds* proposto por Reynolds (1987), Reynolds (1999), e que foi apresentado na Seção 2.1 (Capítulo 2). Assim, a formação *flocking* é alcançada pela modelagem de três leis de interação: alinhamento, aproximação e separação. Itens rotulados desempenham papéis especiais, empurrando para longe outros itens que pertençam a classes distintas. É introduzida também uma técnica de particionamento do espaço cujo propósito é evitar o custo quadrático de se determinar a vizinhança de todos os itens. Argumenta-se ainda que a técnica dinâmica como um todo tem complexidade de tempo sub-quadrática.

Os resultados deste capítulo deram origem ao seguinte artigo:

Roberto A. Gueleri, Thiago H. Cupertino, André C.P.L.F. de Carvalho, and Liang Zhao. **A flocking-like technique to perform semi-supervised learning**. Proceedings of the 2014 International Joint Conference on Neural Networks - IEEE World Congress on Computational Intelligence (IJCNN/WCCI 2014), Beijing, China. Páginas 1579-1586. DOI: 10.1109/IJCNN.2014.6889434

O restante deste capítulo está organizado da seguinte forma. Na Seção 3.1, apresenta-se o modelo dinâmico proposto. A discussão da complexidade computacional e a introdução da técnica de particionamento do espaço acontece na Seção 3.2. Por fim, resultados experimentais com bases de dados artificiais e de *benchmark*, assim como uma comparação com outras técnicas da literatura, são apresentados na Seção 3.3.

Figura 13 – Ilustração da formação *flocking*. Em $t = 0$, o conjunto de dados original. Alguns poucos itens rotulados são representados por símbolos especiais: triângulos e quadrados. Nas demais figuras, o estado do sistema em $t = 100$, $t = 200$ e $t = 300$. Mostra-se a trajetória de todas as partículas pelas últimas 100 iterações. Os eixos horizontal e vertical representam os valores do primeiro e segundo atributos, respectivamente. Figura presente no trabalho de [Gueleri et al. \(2014\)](#).

3.1 Descrição do Modelo

Nesta seção, descreve-se detalhadamente a técnica de aprendizado semissupervisionado baseada em dinâmica coletiva de partículas, ou *flocking*. Começamos, porém, com uma ilustração de como o modelo dinâmico proposto se comporta. Na Figura 13, mostra-se a formação *flocking* num conjunto de dados artificial (conjunto de dados gerado por computador). No início, a cada item de dado (aqui denominado *partícula*) é atribuída uma velocidade aleatória. A magnitude da velocidade permanecerá constante por todo o processo. Apenas sua direção se modificará. Trata-se de uma abordagem comum na modelagem de sistemas *flocking*, como acontece no trabalho de [Vicsek et al. \(1995\)](#). Conforme o processo dinâmico avança, a velocidade de cada partícula vai sendo atualizada de acordo com a localização e a velocidade das partículas ao redor. Elas tendem a se aproximar umas das outras, mas não podem chegar muito perto. Além disso, elas tendem a tomar a mesma direção de movimento. Quanto mais próximas duas partículas estiverem, mais forte será a influência de uma sobre a outra. Itens rotulados desempenham papéis diferenciados, especialmente tratando-se da interação entre dois itens que possuam rótulos distintos: eles tendem a se afastar um do outro. Tal característica promoverá uma competição pela propagação dos rótulos.

Descreve-se agora o modelo dinâmico de maneira precisa. O conjunto de dados é formado pelos seguintes subconjuntos: itens não-rotulados $\mathcal{U}$ e itens rotulados $\mathcal{L}$. Cada item de dado (partícula) $\mathbf{x}_i$ é um vetor no espaço de atributos. N é o tamanho do conjunto de dados (itens não-rotulados mais itens rotulados) e D é a dimensionalidade do espaço de atributos, ou seja, o número de atributos que caracteriza cada item.

Seja $\mathbf{x}_i(t)$ a posição da i -ésima partícula na iteração (tempo) t , e seja $\mathbf{v}_i(t)$ a correspondente velocidade dessa partícula. O movimento dos dados é então definido pelo seguinte conjunto de expressões:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \alpha_1 \cdot \frac{\mathbf{v}_i(t)}{\|\mathbf{v}_i(t)\|} \quad (3.1)$$

$$\mathbf{v}_i(t+1) = \frac{\mathbf{v}_i(t)}{\|\mathbf{v}_i(t)\|} + \alpha_2 \cdot \frac{\mathbf{v}_i^{\text{chg}}(t)}{\|\mathbf{v}_i^{\text{chg}}(t)\|} \quad (3.2)$$

$$\begin{aligned} \mathbf{v}_i^{\text{chg}}(t) = & \sum_{\substack{\mathbf{x}_j | \mathbf{x}_j \in \text{nei}(\mathbf{x}_i, t) \wedge \\ \neg[\mathbf{x}_i, \mathbf{x}_j \in \mathcal{L} \wedge \\ \text{class}(\mathbf{x}_i) \neq \text{class}(\mathbf{x}_j)]}} \text{type}(\mathbf{x}_i, \mathbf{x}_j) \cdot \text{balance}(\mathbf{x}_i, \mathbf{x}_j) \cdot \left[\right. \\ & \text{align} \cdot \frac{\mathbf{v}_j(t)}{\|\mathbf{v}_j(t)\|} \cdot \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^{\beta_1}} + \\ & + \text{approx} \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \cdot \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^{\beta_2}} + \\ & \left. + \text{sep} \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \cdot \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^{\beta_3}} \right] + \\ & + \sum_{\substack{\mathbf{x}_j | \mathbf{x}_i, \mathbf{x}_j \in \mathcal{L} \wedge \\ \text{class}(\mathbf{x}_i) \neq \text{class}(\mathbf{x}_j)}} \text{type}(\mathbf{x}_i, \mathbf{x}_j) \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \quad (3.3) \end{aligned}$$

Embora omitido para não sobrecarregar a Equação 3.3, todos os vetores $\mathbf{x}_i$ e $\mathbf{x}_j$ estão em função de t , ou seja, representam $\mathbf{x}_i(t)$ e $\mathbf{x}_j(t)$, afinal as partículas estão sempre se movimentando.

Pode-se notar que a velocidade é sempre usada como um vetor unitário ($\mathbf{v}_i(t)/\|\mathbf{v}_i(t)\|$). Na Equação 3.1, isso reflete a velocidade de magnitude constante. A Equação 3.3 é apenas uma equação auxiliar que fornece $\mathbf{v}_i^{\text{chg}}(t)$, a alteração (*change*) da velocidade da i -ésima partícula na iteração t . Seu valor é também normalizado quando usado na Equação 3.2 ($\mathbf{v}_i^{\text{chg}}(t)/\|\mathbf{v}_i^{\text{chg}}(t)\|$). Essa última normalização torna seu uso independente do tamanho do conjunto de dados, uma vez que tal tamanho afeta diretamente o valor resultante de $\mathbf{v}_i^{\text{chg}}$.

A segunda somatória na Equação 3.3 ocorre sobre todos os pares $(\mathbf{x}_i, \mathbf{x}_j)$ tal que ambos sejam rotulados e pertençam a classes diferentes. Isso força partículas de classes diferentes a tomarem direções também distintas. A primeira somatória, por sua vez, define o comportamento

flocking propriamente dito e atua sobre todos os demais pares $(\mathbf{x}_i, \mathbf{x}_j)$, exceto aqueles em que ambos $\mathbf{x}_i$ e $\mathbf{x}_j$ são não-rotulados e $\mathbf{x}_j$ não pertence à vizinhança de $\mathbf{x}_i$. Essa filtragem é denotada pela função $\text{nei}(\mathbf{x}_i, t)$, que significa a vizinhança (*neighborhood*) de $\mathbf{x}_i$ na iteração t . Importante salientar que tal filtragem só é aplicada quando ambos $\mathbf{x}_i$ e $\mathbf{x}_j$ são itens não-rotulados. Partículas rotuladas têm “maior importância” e devem sempre ser consideradas, independentemente do quão distantes elas estejam. Uma vez que os itens não-rotulados são encontrados em quantidades muito maiores que os rotulados, e uma vez que os vizinhos mais próximos são os que dominam o valor resultante de $\mathbf{v}_i^{\text{chg}}$, a filtragem pelos vizinhos acaba então reduzindo drasticamente o custo computacional sem desviar-se demais do resultado ideal que se teria em considerar a interação entre todos os pares de itens possíveis. Todos os experimentos apresentados neste capítulo levam em conta somente os cinco vizinhos mais próximos. Na Seção 3.2, uma técnica de particionamento do espaço é apresentada, de modo a evitar-se o custo N^2 de determinar os vizinhos mais próximos a cada iteração dinâmica.

Dentro da primeira somatória da Equação 3.3, há três termos em destaque que são liderados pelos coeficientes *align*, *approx* e *sep*. Como os nomes sugerem, esses termos dizem respeito ao comportamento *flocking* pela combinação de três componentes: alinhamento, aproximação e separação (*alignment*, *approximation* e *separation*). Os coeficientes *align*, *approx* e *sep* são apenas parâmetros do modelo. O termo $(\mathbf{x}_j - \mathbf{x}_i) / \|\mathbf{x}_j - \mathbf{x}_i\|$ é um vetor unitário que fornece a direção de $\mathbf{x}_i$ até $\mathbf{x}_j$. Compare-o com $(\mathbf{x}_i - \mathbf{x}_j) / \|\mathbf{x}_i - \mathbf{x}_j\|$, que fornece o sentido contrário. O termo $1 / \|\mathbf{x}_i - \mathbf{x}_j\|^*$, onde $*$ assume β_1 , β_2 ou β_3 , resulta num fator que depende da distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$. Quanto mais perto estão um do outro, mais $\mathbf{x}_j$ influencia o movimento de $\mathbf{x}_i$. Os símbolos β_1 , β_2 e β_3 são parâmetros. Todos precisam ser positivos, e β_3 deve ser maior que β_2 para que se obtenha o devido equilíbrio entre aproximação e separação. Em todos os experimentos apresentados neste capítulo, atribuem-se $\beta_1 = \beta_2$ e $\beta_3 = \beta_2 + 1$.

O modelo dinâmico aqui apresentado é caracterizado pela interação entre pares de partículas. Cada par pode ser distinguido pelo tipo de seus itens. Assim, é possível fazer um item rotulado exercer maior influência que um não-rotulado, por exemplo. A cada par $(\mathbf{x}_i, \mathbf{x}_j)$ é atribuído então um fator que quantifica a influência de $\mathbf{x}_j$ sobre o movimento de $\mathbf{x}_i$. Esse fator é denotado pela função $\text{type}(\mathbf{x}_i, \mathbf{x}_j)$ (Equação 3.3), que associa valores de acordo com a Tabela 1. Evidentemente, pode-se modificar esses valores a fim de se obter comportamentos dinâmicos diferentes. Os valores mostrados na Tabela 1 são apenas os valores utilizados em nossos experimentos.

Os conjuntos de dados são frequentemente provenientes de classes desbalanceadas, não somente em termos de tamanho, mas também com relação ao número de itens rotulados. A classe detentora da maioria dos itens rotulados tende a absorver também a maioria dos itens não-rotulados. Isso acontece porque cada item rotulado aplica uma força em todos os demais não-rotulados, conseqüentemente fazendo a classe majoritária ser também a mais forte. A fim de evitar esse efeito, necessita-se de um artifício para equilibrar essas forças. É justamente o que

Tabela 1 – Valores atribuídos por $\text{type}(\mathbf{x}_i, \mathbf{x}_j)$.

$\mathbf{x}_i$	$\mathbf{x}_j$	Valor usado nos experimentos
não-rotulado	não-rotulado	1
não-rotulado	rotulado	20, 50 (*)
rotulado	não-rotulado	0.05
rotulado	rotulado (mesma classe)	10
rotulado	rotulado (classes distintas)	10

(*) Valores usados nos conjuntos de dados artificial e de *benchmark*, respectivamente.

faz a função $\text{balance}(\mathbf{x}_i, \mathbf{x}_j)$, na Equação 3.3. Pelos experimentos em conjuntos de duas classes que são apresentados neste capítulo, $\text{balance}(\mathbf{x}_i, \mathbf{x}_j)$ é definido como:

$$\text{balance}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} |\mathcal{L}_1|/|\mathcal{L}_2| & \text{se } \mathbf{x}_j \in \mathcal{L}_2 \wedge \mathbf{x}_i \in \mathcal{U} \\ |\mathcal{L}_2|/|\mathcal{L}_1| & \text{se } \mathbf{x}_j \in \mathcal{L}_1 \wedge \mathbf{x}_i \in \mathcal{U} \\ 1 & \text{se } \mathbf{x}_i \in \mathcal{L} \end{cases} \quad (3.4)$$

onde $\mathcal{L}_1$ e $\mathcal{L}_2$ são os conjuntos de itens rotulados pertencentes às classes 1 e 2, respectivamente, e $|\mathcal{L}_1|$ e $|\mathcal{L}_2|$ são seus tamanhos. Quando $\mathbf{x}_i$ é um item rotulado, $\text{balance}(\mathbf{x}_i, \mathbf{x}_j)$ não tem efeito algum. Generalizações podem também ser derivadas a fim de dar suporte a conjuntos de dados que contenham mais de duas classes.

Por fim, há dois parâmetros a serem explicados: α_1 e α_2 (Equações 3.1 e 3.2). Eles podem ser entendidos como sendo a precisão do processo dinâmico: quanto menor seus valores, mais suave será a dinâmica, mas também mais tempo será necessário para se atingir uma configuração estável (com relação aos grupos que emergem). Certo cuidado deve ser tomado ao atribuir seus valores, pois eles não são independentes um do outro. Eles definem a taxa com que a posição e a velocidade das partículas são atualizadas. Se α_1 for, por exemplo, muito maior que α_2 , então as partículas poderão viajar longas distâncias sem praticamente nenhuma alteração em sua direção de movimento, arruinando a possibilidade da formação dos grupos.

3.2 Complexidade Computacional

Nesta seção, discute-se o custo computacional da técnica aqui tratada, especificamente a complexidade de tempo. As estruturas de dados introduzidas nesta seção não requerem espaço (memória) significativo, portanto a complexidade de espaço permanece na ordem de N , o tamanho do conjunto de dados.

A cada iteração t , o processo iria demandar um custo N^2 , não fosse pela filtragem em que calculam-se somente as interações entre os vizinhos mais próximos, ao invés de interagir

todos com todos. Entretanto, o procedimento natural (simples) de se determinar os vizinhos mais próximos já demandaria um custo N^2 por si só. Torna-se necessário, pois, introduzir uma técnica de particionamento do espaço capaz de evitar tal custo quadrático. Por questão de simplificação, não considera-se que partículas rotuladas e não-rotuladas desempenham papéis diferentes. Uma vez que o número de itens rotulados é bem menor que o de não-rotulados, essa simplificação não causa impacto significativo na análise que segue.

O particionamento do espaço de atributos resulta numa hierarquia de regiões, representada por uma árvore. O nó-raiz denota a maior região considerada, suficientemente grande para conter todo o conjunto de partículas, enquanto seus nós-filho denotam sub-regiões que contêm somente um subconjunto das partículas. A árvore termina em nós-folha, cada qual contendo somente uma partícula. Há uma restrição em que duas partículas não podem ocupar o mesmo local (no início do processo, isso seria um item de dado duplicado), do contrário o processo falhará, pois nenhuma partição será capaz de separá-las. Na Figura 14, ilustra-se como o particionamento do espaço se apresenta.

Conforme o processo dinâmico avança, a cada nova iteração toda a árvore tem que ser reconstruída para se adaptar à nova configuração das partículas. A vizinhança das partículas pode mudar, e o processo de encontrar os novos vizinhos pode ser dividido em duas etapas: (1) construir a árvore e (2) percorrer a árvore para encontrar os vizinhos de cada partícula.

A primeira etapa é descrita da seguinte maneira:

1. Começar com uma região no espaço de atributos tal que cada dimensão é limitada e grande o suficiente apenas para cobrir todo o conjunto de partículas. Esse é o nó-raiz e faz referência a todas as partículas.
2. Dividi-la em 2^D sub-regiões, dividindo cada dimensão ao meio. Chamemos cada nova sub-região de um *ortante* (generalização de quadrantes e octantes).
3. Recursivamente dividir (Passo 2) cada ortante sempre que ele cobrir mais de uma partícula.

Cada nó (ortante) deve referenciar todas as partículas que ele cobre. Ele deve também manter o registro de seus limites no espaço.

Uma árvore razoavelmente balanceada terá uma profundidade em torno de $\log_{(2^D)} N$ (a base é 2^D), que é menor que $\log_2 N$. Portanto o custo de construir tal árvore é da ordem de $N \cdot \log_{(2^D)} N$.

Descreve-se agora como encontrar os K vizinhos mais próximos de uma partícula $\mathbf{x}_i$, percorrendo-se a árvore previamente construída:

1. Partindo-se do nó-raiz, descer a árvore até chegar ao nó-folha que cobre $\mathbf{x}_i$. Tal procedimento é relativamente simples, deve-se apenas olhar para os limites de cada ortante e

Figura 14 – Como o particionamento se apresenta num espaço bidimensional. Retângulos (ortantes, neste caso simplesmente quadrantes) representam os nós da árvore. O maior quadrante é grande o suficiente para cobrir todos os itens e, recursivamente, é dividido em quadrantes menores sempre que houver mais de um item dentro dele. Notar que cada item fica dentro de seu próprio quadrante, que pode ser entendido como seu “endereço” na árvore de particionamento. Itens rotulados aparecem em quantidade muito menor e desempenham papel especial, portanto não são considerados no particionamento. Conexões entre cada item e seus cinco vizinhos mais próximos são também mostradas (em verde), como também são mostradas as trajetórias de todos os itens nas últimas iterações (em cinza). Os eixos horizontal e vertical representam os valores do primeiro e segundo atributos, respectivamente. Figura presente no trabalho de [Gueleri *et al.* \(2014\)](#).

checar se $\mathbf{x}_i$ está dentro, repetindo-se o procedimento para os sub-ortantes. Seria possível também ligar $\mathbf{x}_i$ ao seu nó-folha, evitando assim este passo mas consumindo mais memória. Numa árvore razoavelmente balanceada, porém, esse processo de busca tem custo da ordem de $\log_{(2^D)} N$, portanto é bem rápido.

2. Assim que encontrado o nó-folha, subir um nível (nó-pai) e medir a distância entre $\mathbf{x}_i$ e todas as outras partículas $\mathbf{x}_j$ cobertas por esse nó-pai, selecionando-se as K mais próximas. Lembrar que cada nó referencia todas as partículas por ele cobertas, portanto somente um conjunto bem pequeno de partículas deve ser visitado. Se houver menos de K partículas, subir a árvore um nível a mais e realizar o mesmo procedimento.
3. Uma vez selecionadas as K vizinhas $\mathbf{x}_j$, deve-se verificar a possibilidade de haver alguma outra partícula não-visitada, seja ela $\mathbf{x}_k$, tal que $\mathbf{x}_k$ está mais perto de $\mathbf{x}_i$ do que alguma das

$\mathbf{x}_j$. Para tanto, verificar se a distância entre $\mathbf{x}_i$ e a mais distante das vizinhas $\mathbf{x}_j$ é maior que a distância entre $\mathbf{x}_i$ e os limites da região visitada (o ortante visitado de nível mais alto; lembrar que todo nó carrega essa informação). Caso não, garante-se que as vizinhas encontradas são, de fato, as vizinhas mais próximas. Caso sim, porém, torna-se necessário visitar um nó acima e repetir o processo. A determinação precisa do custo computacional deste passo é uma tarefa difícil. Além disso, tal custo depende de D e K . Na média, porém, espera-se que o número de partículas visitadas seja muito menor que o número total N . Pode-se então dizer que o custo de encontrar as vizinhas mais próximas de uma certa partícula tem ordem menor que N (mesmo considerando o Passo 1), consequentemente o custo de encontrar a vizinhança de todo o conjunto de partículas é menor que N^2 .

Juntando-se os dois procedimentos (construir a árvore e percorrê-la), o custo total permanece abaixo de N^2 . Esse é o custo total de encontrar os K vizinhos mais próximos de todos os itens, assumindo que K seja suficientemente pequeno.

Agora deve-se considerar o custo de calcular as interações entre as partículas, como descrito na Seção 3.1, Equações 3.1, 3.2 e 3.3. Somente as somatórias da Equação 3.3 são relevantes para nossa análise, visto que estamos interessados no custo assintótico, ou seja, o custo quando o tamanho N do conjunto de dados cresce indefinidamente. A cada iteração t , cada partícula $\mathbf{x}_i$ interage com suas K vizinhas $\mathbf{x}_j$. Visto que K é uma constante pequena, o custo de calcular as interações entre todas as partículas é da ordem de N , portanto linear. Somado ao custo de determinar os vizinhos, ainda tem-se uma complexidade sub- N^2 .

Finalmente, deve-se notar que toda a análise feita até agora se dá sobre uma única iteração t . A questão então é: quantas iterações seriam necessárias para se obter bons resultados de classificação? Essa é uma questão muito mais intrigante e sua análise completa está além do escopo deste trabalho. Contudo, alguns resultados empíricos foram obtidos.

Na Figura 15, mostra-se como a acurácia de classificação depende do valor de α_1 e do número de iterações t . O parâmetro α_2 foi mantido em 1 para todas as simulações. Os demais parâmetros foram ajustados conforme discutido na seção de experimentos a seguir (Seção 3.3). A cada iteração, a classificação é realizada fazendo-se cada item não-rotulado receber o rótulo do item rotulado mais próximo. Notar que em $t = 0$, o resultado é simplesmente idêntico à aplicação da técnica “1-Nearest Neighbor” (1-NN). Observa-se claramente que valores menores de α_1 (precisão mais fina) conduzem a melhor acurácia de classificação. Pode-se formular a hipótese na qual a causa dos resultados ruins para valores maiores de α_1 reside no início do processo, pois os dados iniciam seu movimento com direções aleatórias, e grandes passos usando direções aleatórias simplesmente degradam o padrão original do conjunto de dados. Tendo isso em mente, avaliou-se também uma variante do modelo, na qual utilizam-se valores incrementais de α_1 , de modo que α_1 começa bem pequeno e, à medida que o sistema vai se estabilizando (com relação à direção das partículas), seu valor vai aumentando. Tais resultados são mostrados no lado direito da Figura 15. Em ambos os conjuntos de dados, obtêm-se resultados tão bons quanto aqueles de

Figura 15 – Como a acurácia de classificação depende do valor de α_1 e do número de iterações t . Em cima, conjunto de dados “duas gaussianas”, ou simplesmente “gaussianas” (Figura 16). Embaixo, conjunto de dados “arcos entrelaçados”, ou simplesmente “arcos” (Figura 16). À esquerda, valor constante de α_1 por todo o processo. Experimentos realizados com cinco diferentes valores de α_1 . Do vermelho ao azul: 0,005, 0,01, 0,02, 0,04 e 0,08 no conjunto das gaussianas (gráfico superior-esquerdo), e 0,01, 0,02, 0,04, 0,08 e 0,16 no conjunto dos arcos (gráfico inferior-esquerdo). Cada linha é a média de 30 simulações, medida a cada 10 iterações. À direita, experimentos utilizando valores incrementais de α_1 . Em ambos os conjuntos de dados (gráficos superior-direito e inferior-direito), α_1 começa valendo 0,001 e é multiplicado por um fator de 1,2 a cada 5 iterações, até atingir um valor máximo de 0,2. Novamente, cada linha é a média de 30 simulações, mas agora medida a cada 5 iterações. Figura presente no trabalho de [Gueleri et al. \(2014\)](#).

pequenos valores de α_1 , mas agora demandando muito menos iterações.

Espera-se também que o número de iterações necessárias para se atingir bons resultados de classificação não cresça com o tamanho N do conjunto de dados, ao menos não cresça linearmente. A dependência que a qualidade de classificação tem de t parece residir na geometria dos dados, e não diretamente em N . Assumindo isso como verdade, pode-se afirmar que o custo assintótico de tempo demandado por todo o processo de classificação continua sendo $\text{sub-}N^2$. Entretanto, a maneira como o número de iterações é influenciado por N ainda merece mais estudo.

3.3 Resultados Experimentais

A fim de verificar a acurácia de classificação da técnica proposta, apresentam-se aqui resultados experimentais realizados em bases de dados tanto artificiais (geradas por computador) quanto de *benchmark*. O ajuste de parâmetros é apresentado primeiramente. O fator $\text{type}(\mathbf{x}_i, \mathbf{x}_j)$ (Equação 3.3, Seção 3.1) é ajustado de acordo com a Tabela 1 (Seção 3.1). O ajuste do parâmetro α_1 depende da densidade dos dados. Os valores utilizados estão na faixa de 0,01 para bases de dados compactas, até 0,04 para as mais esparsas. O parâmetro α_2 foi definido como 1. O parâmetro β_1 foi definido como 2 nas bases artificiais e definido entre 2 e 10 nas bases de *benchmark*. Já os parâmetros β_2 e β_3 foram ajustados como $\beta_2 = \beta_1$ e $\beta_3 = \beta_2 + 1$. Somente os cinco vizinhos mais próximos foram considerados para o cálculo da interação entre itens não-rotulados.

Observando a Equação 3.3, pode-se notar que se os termos de aproximação e de separação forem isolados, haverá uma distância $\|\mathbf{x}_i - \mathbf{x}_j\|$ com a qual esses termos neutralizarão um ao outro. Essa é uma das ideias por trás da formação *flocking*: duas partículas tendem a se aproximar uma da outra, mas não muito. Assim, ao invés de ajustar *approx* e *sep* de modo independente, é melhor estabelecer uma distância de equilíbrio d_{eq} desejada. Assumindo que $\beta_3 = \beta_2 + 1$, então simplesmente define-se $\text{approx} = 1$ e isso resulta que $\text{sep} = d_{eq}$. A fim de tornar o ajuste de parâmetros menos dependente da base de dados, em todas as simulações define-se d_{eq} como sendo a média das distâncias entre cada item não-rotulado e seu vizinho mais próximo. Por fim, o parâmetro *align* é definido como 1.

3.3.1 Bases de Dados Artificiais

Na Figura 16, apresentam-se mapas de classificação em duas bases de dados artificiais. Cada base tem 1000 itens, alguns deles inicialmente rotulados e denotados por símbolos especiais: triângulos e quadrados. Cada cor representa uma classe: vermelho para “triângulo” e azul para “quadrado”. Cada entrada do mapa (pequena região quadrada) é calculada adicionando-se um item não-rotulado exatamente naquele local, executando-se o processo dinâmico e, por fim, verificando-se qual rótulo aquele item adquire. O rótulo que ele adquire é simplesmente o rótulo do item rotulado mais próximo, similar à classificação 1-NN, mas medida após certo número de iterações dinâmicas (750 iterações para as gaussianas e 1000 iterações para os arcos). Devido à atribuição inicialmente aleatória da direção das partículas, cada entrada do mapa é a média de 15 simulações completas. Valores intermediários entre o vermelho e o azul significam que a partícula adicional terminou sendo classificada ora como da classe vermelha, ora como da classe azul. Definiu-se $\alpha_1 = 0,01$ para as gaussianas e $\alpha_1 = 0,02$ para os arcos. Apenas para comparação, são mostrados também os mapas de classificação obtidos com o algoritmo 1-NN.

Embora o 1-NN seja muito mais rápido, os resultados obtidos com a técnica baseada em *flocking* são muito mais consistentes. Além disso, os mapas de classificação mostram regiões

Figura 16 – Mapas de classificação em duas bases de dados artificiais: “duas gaussianas”, ou simplesmente “gaussianas” (à esquerda) e “arcos entrelaçados”, ou simplesmente “arcos” (à direita). Em cima, emprego da técnica proposta. Cada pequeno quadrado do mapa é resultado da média de 15 simulações, cada uma medida após 750 iterações nas gaussianas, e 1000 iterações nos arcos. Embaixo, uso do 1-NN. Os eixos horizontal e vertical representam os valores do primeiro e segundo atributos, respectivamente. Figura presente no trabalho de [Gueleri et al. \(2014\)](#).

de difícil rotulação, onde aparecem cores entre o vermelho e o azul. Essa propriedade pode ser explorada para rotulação *soft*, ou *fuzzy*, ao invés de simplesmente atribuir-se um único rótulo a cada item. Em outras palavras, cada item a ser classificado pode receber um grau de pertinência a cada uma das classes. Na prática, considerando uma tarefa de aprendizado transdutivo, tais mapas nem são necessários para uma classificação *soft*. Apenas deve-se executar o sistema dinâmico um certo número de vezes, usando inicializações aleatórias diferentes para a direção das partículas, e então comparar o número de vezes que um certo item não-rotulado acaba classificado como uma classe ou outra.

3.3.2 Bases de Benchmark

A fim de comparar a performance da técnica aqui proposta com outras técnicas bem conhecidas da literatura, foram conduzidos também experimentos em bases de *benchmark*. Na Tabela 2, são listadas as bases utilizadas, como proposto por [Chapelle, Schölkopf e Zien \(2006\)](#). Cada base de dados está disponível em 12 versões diferentes, chamadas de *splits*. Os itens de dados são exatamente os mesmos, mas cada *split* possui um conjunto diferente de itens

Tabela 2 – As bases de *benchmark* utilizadas nos experimentos, como proposto por [Chapelle, Schölkopf e Zien \(2006\)](#).

Base de dados	Classes	Dimensões (*)	Tamanho
g241c	2	241	1500
g241d	2	241	1500
Digit1	2	241	1500
USPS	2	241	1500
BCI	2	117	400

(*) Todos os resultados apresentados neste trabalho, incluindo a aplicação das outras técnicas da literatura, foram obtidos utilizando-se somente as cinco primeiras componentes PCA (Principal Component Analysis).

Tabela 3 – Referências das outras técnicas usadas para comparação.

Abreviação	Técnica	Referência
k-NN	k-Nearest Neighbors	(COVER; HART, 1967)
MVU	Maximum Variance Unfolding	(SUN <i>et al.</i>, 2006)
LEM	Laplacian Eigenmaps	(BELKIN; NIYOGI, 2003 ; BELKIN; NIYOGI; SINDHWANI, 2006 ; BELKIN; NIYOGI, 2008 ; ROSENBERG <i>et al.</i>, 2009 ; MINH; SINDHWANI, 2011)
SVM	Support Vector Machines	(CORTES; VAPNIK, 1995)
TSVM	Transductive Support Vector Machines	(CHAPELLE; ZIEN, 2005 ; JOACHIMS, 2006 ; CHAPELLE; SINDHWANI; KEERTHI, 2008)
SGT	Spectral Graph Transducer	(JOACHIMS, 2003)
LDS	Low-Density Separation	(CHAPELLE; ZIEN, 2005)
Lapl. RLS	Laplacian Regularized Least Squares	(SINDHWANI; NIYOGI; BELKIN, 2005 ; BELKIN; NIYOGI; SINDHWANI, 2006 ; ROSENBERG <i>et al.</i>, 2009 ; MINH; SINDHWANI, 2011)
Attr. Forces	Attraction Forces	(CUPERTINO; GUELERI; ZHAO, 2014)

pré-rotulados.

Na Tabela 3, apresentam-se as técnicas obtidas da literatura, que foram empregadas para comparação da acurácia de classificação. A correspondente referência de cada técnica é também fornecida.

A comparação da acurácia de classificação obtida entre as diferentes técnicas é apresentada na Tabela 4. As bases foram pré-processadas, de modo que somente as cinco primeiras componentes PCA (Principal Component Analysis) foram utilizadas nos experimentos, com exceção das técnicas MVU e LEM, que são técnicas de redução de dimensionalidade. Os resulta-

Tabela 4 – Erros de classificação (%) nas bases de dados propostas por Chapelle, Schölkopf e Zien (2006).

As bases foram pré-processadas, de modo que somente as cinco primeiras componentes PCA foram utilizadas nos experimentos, com exceção das técnicas MVU e LEM, que são técnicas de redução de dimensionalidade. Os resultados correspondentes às técnicas da literatura foram obtidos de Cupertino, Gueleri e Zhao (2014). Com relação à nossa técnica, diferentes combinações de β_1 , β_2 e β_3 foram avaliadas. Em todas elas, β_2 e β_3 foram definidas como $\beta_2 = \beta_1$ e $\beta_3 = \beta_2 + 1$. Os melhores resultados da literatura são realçados, como também são realçados os melhores resultados da nossa técnica. A posição da nossa técnica no *ranking* (entre 13) é também mostrada. Entradas marcadas com “-” não foram simuladas.

	10 itens pré-rotulados					100 itens pré-rotulados				
	g241c	g241d	Digit1	USPS	BCI	g241c	g241d	Digit1	USPS	BCI
1-NN	28.95	27.20	21.77	23.93	49.10	21.40	10.98	7.04	13.60	47.36
MVU + 1-NN	47.15	45.56	14.42	23.34	47.95	43.01	38.20	2.83	6.50	47.89
LEM + 1-NN	44.05	43.22	23.47	19.82	48.74	40.28	37.49	6.12	7.64	44.83
SVM (linear)	24.00	28.67	21.53	31.52	48.50	13.34	13.99	6.79	21.60	46.92
SVM (RBF)	44.41	39.23	33.15	19.69	49.79	17.76	9.12	5.73	13.68	48.06
SVM (quad)	39.33	31.95	26.62	25.62	49.64	17.56	10.41	6.18	12.88	49.00
SVM (poly)	33.47	33.23	29.63	26.09	49.44	22.05	13.48	7.40	18.45	47.75
TSVM	21.38	31.23	20.32	30.04	48.96	13.77	21.63	9.15	14.54	45.65
SGT	17.83	12.94	14.63	26.50	49.64	14.11	6.72	4.91	13.57	48.50
LDS	17.46	36.48	18.90	23.48	49.02	13.19	8.39	4.49	19.33	47.31
Laplacian RLS	27.10	26.67	18.03	18.68	49.38	16.04	7.83	4.88	11.76	46.11
Attr. Forces	27.64	27.42	18.31	18.31	49.64	21.63	10.24	6.41	12.44	45.63
Nossa técnica	35.08	34.70	23.84	70.21	49.90	20.02	12.47	8.11	78.99	48.64 $\beta_1 = 2$
	29.48	29.80	22.57	55.15	49.81	17.24	9.43	6.44	58.04	47.99 $\beta_1 = 3$
	27.57	27.98	21.88	46.44	49.87	17.17	8.80	6.20	40.70	47.69 $\beta_1 = 4$
	27.00	27.72	23.37	39.29	49.71	17.51	8.75	6.41	30.27	47.50 $\beta_1 = 5$
	27.24	28.16	26.28	35.72	49.75	17.90	8.97	6.98	25.14	47.38 $\beta_1 = 6$
	-	-	-	35.52	49.72	-	-	-	20.43	47.54 $\beta_1 = 8$
	-	-	-	36.75	49.86	-	-	-	19.48	47.03 $\beta_1 = 10$
Posição (/13)	5	5	9	13	12	6	4	8	12	6

dos correspondentes às técnicas da literatura foram obtidos de [Cupertino, Gueleri e Zhao \(2014\)](#). Resultados nas bases de dados originais, sem pré-processamento para obtenção das componentes PCA, podem ser encontrados no livro de [Chapelle, Schölkopf e Zien \(2006\)](#). Definiram-se $\alpha_1 = 0,01$ para a base *Digit1*, $\alpha_1 = 0,02$ para *g241c*, *g241d* e *USPS*, e $\alpha_1 = 0,04$ para *BCI*. As medidas foram obtidas após 750 iterações. Cada entrada na tabela (para nossa técnica) é a média de $12 \text{ splits} \times 10$ execuções por *split*.

A partir desses experimentos, é possível ver que a nossa técnica baseada em *flocking* apresenta bom desempenho nas bases *g241c* e *g241d*, desempenho mediano na *Digit1*, e ruim nas *USPS* e *BCI*. Na verdade, todos os algoritmos apresentam desempenho ruim na base *BCI*. Os melhores resultados da literatura têm taxas de erro de 47,95 e 44,83% (10 e 100 itens pré-rotulados, respectivamente). Uma simples atribuição aleatória resultaria numa taxa de erro em torno de 50%, quase a mesma taxa obtida por todos os algoritmos.

Estranhamente, nossa técnica resulta numa taxa de erro muito acima de 50% para valores baixos de β_1 , na base *USPS*. Nota-se claramente que a acurácia depende bastante de β_1 , mas mesmo para maiores valores, resultados ruins ainda são obtidos. Na verdade, a dependência do valor de β_1 pode ser notada em todas as bases de dados.

As bases *g241c*, *g241d* e *Digit1* seguem, respectivamente, os três pressupostos (*assumptions*) discutidos por [Chapelle, Schölkopf e Zien \(2006\)](#), a saber: *cluster*, *smoothness* e *manifold*. Assim, esses experimentos sugerem que a técnica aqui proposta tem boa atuação nos pressupostos de *cluster* e *smoothness*.

O método k-NN é muito rápido, tendo complexidade de tempo linear, assumindo que a quantidade de itens rotulados é bem menor que a de não-rotulados. Entretanto, ele é tão simples que geralmente conduz a classificação equivocada, como mostrado na Figura 16. Na verdade, ele nem leva em conta o padrão de distribuição dos itens não-rotulados. A maioria dos outros métodos, por sua vez, apresentam complexidade de tempo no mínimo quadrática. Os métodos baseados em grafo, por exemplo, normalmente utilizam uma matriz de adjacência, o que por si só já conduz a um custo quadrático. Como discutido na seção de análise de complexidade (Seção 3.2), nossa técnica apresenta complexidade de tempo sub-quadrática (do ponto de vista assintótico, para bases muito grandes), portanto atinge-se um equilíbrio entre boa acurácia de classificação e custo computacional aceitável.

SEGMENTAÇÃO DE IMAGENS VIA DINÂMICA COLETIVA DE SUPERPIXELS

Segmentação de imagens é o processo de particionar uma imagem em segmentos distintos, agrupando-se pixels que compartilham propriedades similares. Em outras palavras, a segmentação simplifica a representação da imagem, atribuindo-se um rótulo a cada pixel. Desse modo, processamentos subsequentes não mais precisam “olhar” para a imagem bruta, mas apenas para seu conjunto de segmentos, algo que confere à segmentação de imagens um papel fundamental no campo da visão computacional. O objetivo da segmentação de imagens é, portanto, fornecer uma separação dos diferentes objetos que compõem uma cena, por exemplo, isolar um objeto do plano de fundo.

Diversas abordagens já foram propostas na literatura de segmentação de imagens. Por exemplo, métodos baseados em histograma levam em conta os picos e vales de um histograma para decidir onde separar a imagem (OHLANDER; PRICE; REDDY, 1978; TOBIAS; SEARA, 2002). Eles geralmente utilizam a cor ou a intensidade dos pixels para construir o histograma, e tipicamente não consideram a distribuição espacial dos pixels. Esses métodos costumam ser muito rápidos, mesmo quando requerem mais de um passo por meio do refinamento recursivo das regiões segmentadas (OHLANDER; PRICE; REDDY, 1978). Outro exemplo são os métodos baseados em expansão de regiões, que começam com pequenas regiões — ou mesmo pixels individuais — e iterativamente funde regiões vizinhas de acordo com algum critério (ADAMS; BISCHOF, 1994; NOCK; NIELSEN, 2004; SHIH; CHENG, 2005; ALPERT *et al.*, 2012). Outra abordagem comum para segmentação de imagens é encontrada em métodos baseados em grafo (FELZENSZWALB; HUTTENLOCHER, 2004; GRADY, 2006; COUPRIE *et al.*, 2011; PENG; ZHANG; ZHANG, 2013; WANG *et al.*, 2015). Eles modelam a imagem como sendo um grafo, fazendo cada pixel — ou cada pequena região de pixels — ser um vértice do grafo, e fazendo cada aresta conectar pixels vizinhos e carregar um valor de peso que define a similaridade entre um par de pixels.

Neste capítulo, é apresentada uma técnica de segmentação de imagens baseada no movimento de superpixels por um espaço multidimensional. Essa abordagem toma como inspiração a formação *flocking* descentralizada e auto-organizável que pode ser vista na natureza, combinando os modelos de localização e velocidade, e que forma a base deste trabalho de doutorado. Faz-se cada superpixel ser um indivíduo que se move pelo espaço, denotado aqui como sendo uma *partícula*. Partindo com direções aleatórias, cada partícula se move em conformidade com sua vizinhança, tentando tomar a mesma direção de movimento. A influência entre um par de partículas será tão maior quanto mais próximas estiverem uma da outra. Ao final do processo, grupos — os segmentos — são estabelecidos entre aqueles superpixels que tomaram a mesma direção de movimento, e que consequentemente mantiveram-se próximos uns dos outros. Apresenta-se também uma ampla análise de parâmetros, medindo a qualidade dos resultados com imagens de referência previamente anotadas. É também realizada a comparação com algumas outras técnicas de segmentação, o que nos permite discutir alguns prós e contras da abordagem baseada na dinâmica coletiva (*flocking*) de superpixels.

Os resultados deste capítulo deram origem aos seguintes trabalhos:

Roberto Alves Gueleri and Liang Zhao. **Image segmentation via superpixels self-organized motion**. Proceedings of the 4th Symposium on Knowledge Discovery, Mining and Learning (KDMILE 2016), Recife, PE. Páginas 44-51.

Roberto Alves Gueleri, Qiusheng Zheng, Junbao Zhang and Liang Zhao. **Image segmentation via flocking-like particle dynamics**. Aceito para inclusão no livro “Swarm intelligence: from concepts to applications”, editado por Ying Tan. Editora IET (UK).

O restante deste capítulo está organizado da seguinte forma. Na Seção 4.1, descreve-se detalhadamente o modelo dinâmico proposto. Na Seção 4.2, apresenta-se primeiramente uma visão geral dos resultados que podem ser obtidos com a técnica de segmentação baseada em *flocking*. Após isso, apresenta-se uma análise detalhada de parâmetros, seguida pela comparação com alguns outros algoritmos.

4.1 Descrição do Modelo

Dada uma imagem, encontra-se primeiramente um conjunto de superpixels capaz de representá-la adequadamente. Um *superpixel* representa um pequeno grupo de pixels da imagem, tipicamente pixels próximos uns dos outros e com características semelhantes, como cor, brilho etc. Fazer cada pixel da imagem representar diretamente uma partícula do sistema dinâmico o tornaria computacionalmente muito custoso. Eis a motivação do uso dos superpixels: um sistema contendo uma quantidade moderada de partículas, e cada partícula sendo um bom representante

de uma pequena região da imagem. Esse pré-processamento pode ser realizado com o auxílio de técnicas de agrupamento. Nos experimentos aqui realizados, o algoritmo k-médias é utilizado para determinar o conjunto de superpixels. Técnicas um pouco mais sofisticadas poderiam ainda ser empregadas, por exemplo o algoritmo SLIC, que é também baseado no k-médias e próprio para a obtenção de superpixels ([ACHANTA *et al.*, 2012](#)).

Após determinar o conjunto de superpixels, cada um é representado por uma partícula motora, que é composta por cinco ou mais dimensões:

- As duas primeiras são as dimensões horizontal e vertical da imagem.
- As três seguintes descrevem os atributos de cor da imagem. O modelo de cor RGB é considerado neste trabalho.
- Dimensões adicionais reduzem a chance de colisão entre grupos distintos de partículas, mas tornam o método mais custoso. Uma dimensão adicional é utilizada nos experimentos deste trabalho, totalizando seis dimensões. Mais à frente, na seção de análise de parâmetros (Seção 4.2.2), mostra-se um exemplo em que dois grupos distintos colidem um com o outro (Figuras 29, 30 e 31).

O processo de segmentação consiste então nas seguintes etapas:

1. Dado um valor máximo desejado de superpixels, a localização inicial de seus centroides é disposta formando uma grade regular sobre a imagem. As distâncias horizontal e vertical que separam centroides adjacentes devem ser as mais semelhantes possíveis. A quantidade definitiva de superpixels será a máxima capaz de formar a grade regular, mas sem exceder o valor máximo desejado. Com exceção das duas primeiras dimensões, todos os centroides terão coordenada inicial nula. Os dados são normalizados de modo que as componentes de cor variem no intervalo $[-0,5; +0,5]$ e que as distâncias horizontal e vertical entre centroides adjacentes também sejam unitárias, ou o mais próximo disso, visto que essas duas distâncias são parecidas mas não necessariamente iguais. Precisamente, faz-se a média entre as distâncias horizontal e vertical valer 1.
2. O algoritmo k-médias é aplicado para ajustar a posição dos centroides. Ao final, valores não-nulos serão obtidos para as três dimensões de cor. Esse é um passo fundamental, pois ele define a disposição inicial do sistema dinâmico, e conseqüentemente exerce forte influência no resultado final de segmentação. Lembrar que cada centroide representa um superpixel, que por sua vez representa um grupo de pixels. Portanto, até o final do processo, haverá um vínculo entre cada centroide e os pixels da imagem que ele representa. Cada centroide será tratado como uma partícula $\mathbf{x}_i(t)$ do sistema dinâmico, onde o índice de iteração t (*tempo*) começa valendo zero, representando a configuração inicial do sistema.

3. As partículas são colocadas a se movimentar. Mas antes, uma renormalização dos dados é necessária, fazendo com que as componentes de cor sejam expandidas para que seu intervalo, que era de $[-0,5; +0,5]$, agora se iguale ao tamanho da imagem, em número de superpixels. Similarmente ao que foi feito antes, toma-se a média entre os tamanhos horizontal e vertical. Tal procedimento atribui um mesmo grau de “importância” entre coordenada na imagem e valor de cor. As expressões que regem a movimentação das partículas são apresentadas logo à frente, ainda nesta seção. Uma condição de parada é também necessária, e para isso define-se o conceito de *raio*, que é maior das distâncias (euclidiana) entre a posição de uma partícula e a origem do sistema de coordenadas do espaço. Para tanto, no início do processo, é importante que o conjunto de partículas esteja centralizado em torno da origem. Nos experimentos conduzidos neste trabalho, o processo é interrompido quando atinge-se um raio 16 vezes maior que o raio inicial. Espera-se que os dados já estejam bem separados e que padrões úteis já tenham surgido nesse momento.

Precisamente, a movimentação de cada partícula $\mathbf{x}_i$ é regida pelo seguinte sistema:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \alpha_1(t) \cdot \hat{\mathbf{v}}_i(t) \quad (4.1)$$

$$\mathbf{v}_i(t+1) = \hat{\mathbf{v}}_i(t) + \alpha_2 \cdot \hat{\mathbf{v}}_i^{\text{chg}}(t) \quad (4.2)$$

$$\mathbf{v}_i^{\text{chg}} = \sum_{\mathbf{x}_j \in \text{nei}(\mathbf{x}_i)} (\hat{\mathbf{v}}_j - \hat{\mathbf{v}}_i) \cdot \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^\beta}. \quad (4.3)$$

A velocidade inicial $\mathbf{v}_i(0)$ de todas as partículas é atribuída aleatoriamente. A localização inicial $\mathbf{x}_i(0)$ das partículas é a disposição dos centroides (superpixels), formando uma grade sobre a imagem. Embora omitido para simplificação, como ocorre na Equação 4.3, todas as variáveis $\mathbf{x}$, $\mathbf{v}$ e $\mathbf{v}^{\text{chg}}$ são sempre em função de t .

Os símbolos $\hat{\mathbf{v}}_i$ e $\hat{\mathbf{v}}_i^{\text{chg}}$ denotam vetores unitários, isto é, $\hat{\mathbf{v}}_i = \mathbf{v}_i / \|\mathbf{v}_i\|$ e $\hat{\mathbf{v}}_i^{\text{chg}} = \mathbf{v}_i^{\text{chg}} / \|\mathbf{v}_i^{\text{chg}}\|$. Especialmente, na Equação 4.1, $\hat{\mathbf{v}}_i$ força todas as partículas a desempenharem a mesma velocidade (magnitude). $\|\mathbf{x}_i - \mathbf{x}_j\|$ é a distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$. A distância euclidiana é empregada neste trabalho.

O termo $\text{nei}(\mathbf{x}_i)$ representa a vizinhança (*neighborhood*) de $\mathbf{x}_i$. Essa vizinhança também é uma função de t e deve ser atualizada regularmente, conforme as partículas se movem. Fazer a interação ocorrer somente entre os vizinhos mais próximos, ao invés de todos interagirem com todos, é necessário para manter viável o custo computacional. Neste trabalho, cada partícula interage com seus oito vizinhos mais próximos. Importante notar que uma técnica de particionamento do espaço é utilizada a fim de evitar o custo quadrático que se teria em determinar quais são os vizinhos mais próximos. Trata-se da técnica descrita no Capítulo 3. A atualização dos vizinhos é realizada a cada 50 iterações.

Figura 17 – Ilustração do processo de segmentação. (a): Imagem original. (b): Imagem reduzida em superpixels. (c): Configuração inicial do sistema, em que cada superpixel é uma partícula dinâmica. Os vetores-velocidade apontam para direções aleatórias (segmentos de reta vermelhos). Neste exemplo, seis dimensões são utilizadas: dimensões horizontal e vertical da imagem, mais três componentes de cor (RGB), mais uma dimensão adicional para evitar colisões. Somente a projeção nas duas primeiras dimensões é mostrada. (d, e): Após algum tempo. (f): Instante em que o processo dinâmico é interrompido e as partículas são agrupadas. (g): O resultado da obtenção de dois segmentos, usando-se o algoritmo k-médias. Figura presente no trabalho de [Gueleri *et al.* \(2018\)](#).

Figura 18 – Imagens empregadas nos experimentos. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Os parâmetros α_1 e α_2 definem a intensidade com que a posição e a velocidade das partículas variam, respectivamente. Notar que α_1 é uma função de t , enquanto α_2 é constante por todo o processo. Uma vez que os vetores-velocidade apontam para direções aleatórias no início, é conveniente deixar primeiro essas velocidades se acomodarem, para só então permitir que as partículas desempenhem um movimento mais intenso. Para tanto, α_1 é incrementado gradativamente com o avanço de t . Precisamente,

$$\alpha_1 = 10^{\exp} \cdot t^3, \quad (4.4)$$

tal que diferentes valores do expoente $\exp$ levam a comportamentos distintos, e são analisados neste capítulo. Valores menores de expoente oferecem mais tempo para que as partículas se alinhem, assim o movimento só se torna visível após esse alinhamento. Uma ilustração do processo de segmentação é apresentada na Figura 17.

4.2 Resultados Experimentais

Nesta seção, são apresentados e discutidos resultados empíricos, que são obtidos com a técnica de segmentação baseada na dinâmica coletiva de superpixels. Começamos com uma visão geral dos resultados na Seção 4.2.1, seguida de uma análise detalhada de parâmetros na Seção 4.2.2, e por fim uma comparação com alguns outros algoritmos na Seção 4.2.3.

4.2.1 Visão Geral dos Resultados

O conjunto de imagens utilizadas em nossos experimentos é mostrado na Figura 18. A primeira é uma imagem artificial (“estrela”). Essa imagem impõe algumas dificuldades, especialmente nas quatro pontas da estrela, que tendem a exibir a mesma cor do gradiente de fundo. A segunda imagem é uma fotografia (“lago”). Embora o céu esteja claro e o lago exiba uma cor verde bem vívida, as construções ao meio têm uma variedade de formas e cores. Sombras

também aparecem por toda a imagem, e o céu ensolarado pode facilmente ser confundido com algumas outras áreas claras. A última imagem é também uma fotografia (“torre”), na qual o céu de fundo é facilmente distinguível das construções à frente. Entretanto, essas construções possuem cores e intensidades similares, o que torna difícil a tarefa de isolá-las umas das outras. Além disso, o céu aparece levemente nublado e espacialmente descontínuo devido à torre, algo que tende a quebrá-lo em mais de um segmento. Para todo experimento apresentado neste capítulo, inclusive na comparação com outros algoritmos, os tamanhos das imagens de entrada são: 800x800 (estrela), 1200x675 (lago) e 750x1000 (torre).

Exemplos de resultados que podem ser alcançados com a técnica baseada em *flocking* são apresentados na Figura 19. Com relação à imagem de estrela, uma deficiência aparece ao longo das bordas da forma de estrela. Tais irregularidades são decorrentes da forma que alguns superpixels adquirem em regiões de fronteira que possuem cores similares. Em outras palavras, isso é causado pelo modo como os superpixels são criados, algo que não tem nenhuma relação com a dinâmica coletiva discutida neste trabalho, exceto por definir a disposição inicial das partículas. Métodos mais sofisticados podem também ser empregados na obtenção dos superpixels, por exemplo, o algoritmo SLIC (ACHANTA *et al.*, 2012), o que poderia definir melhor as bordas entre diferentes segmentos. Na Figura 20, mostra-se como os superpixels aparecem quando se aplica o método descrito anteriormente na Seção 4.1. Apesar de algumas irregularidades, o método utilizado funciona bem em regiões de bordas nítidas. Talvez seja óbvio, mas a obtenção de um maior número de superpixels torna possível capturar pequenos detalhes da cena. Por exemplo, a pequena antena no topo da torre é melhor capturada ao se utilizar 2000 superpixels.

Observando novamente a Figura 19, apesar dos erros de classificação em algumas pequenas regiões, é possível ver que os resultados de segmentação como um todo são consistentes. Perdem-se as finas pontas da estrela, mas a estrela como um todo é corretamente destacada do gradiente de fundo. Mais importante ainda, todo o plano de fundo é mantido como um único segmento. Com relação à imagem do lago, perdem-se novamente alguns pequenos detalhes, mas a imagem como um todo é apropriadamente particionada em três grandes segmentos, incluindo a região heterogênea que contém as construções. Finalmente, com relação à imagem da torre, não apenas isola-se o céu de fundo como um único e grande segmento, mas também divide-se corretamente o plano de frente em três construções distintas.

Observando a disposição final das partículas, vale enfatizar que cada segmento é definido por grupos claramente distintos. Nesses casos, torna-se fácil determinar os segmentos da imagem a partir da disposição que as partículas tomam, diferentemente da configuração inicial que se tem antes da dinâmica coletiva das partículas entrar em ação.

Por fim, apenas para que se tenha uma visão geral do efeito causado por diferentes parâmetros, um painel com diferentes resultados para uma mesma imagem (torre) é mostrado na Figura 21. A análise detalhada dos parâmetros é, contudo, apresentada na seção seguinte.

Figura 19 – Exemplos de resultados obtidos com a técnica baseada na dinâmica coletiva de superpixels. À esquerda, diferentes segmentos realçados por cores distintas. À direita, a disposição final das partículas, em duas projeções (1^a vs. 2^a, e 1^a vs. 3^a dimensões). As partículas são pintadas de acordo com os grupos encontrados pelo algoritmo k-médias, o que acaba conduzindo aos segmentos mostrados à esquerda. O parâmetro de expoente vale -14 , -16 e -15 , para as imagens de estrela, lago e torre, respectivamente. Todas essas simulações são realizadas com 2000 superpixels. Demais parâmetros: $\alpha_2 = 0,2$ and $\beta = 2$. Figura presente no trabalho de [Gueleri *et al.* \(2018\)](#).

Figura 20 – Efeitos da obtenção de diferentes quantidades de superpixels. Da esquerda para a direita, 500, 1000 e 2000 superpixels. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

4.2.2 Análise de Parâmetros

A fim de entender a influência de diferentes configurações de parâmetros na segmentação final, resumam-se os resultados de uma análise ampla de parâmetros pelas Figuras 22, 23, 24 e 25.

Um índice de avaliação é utilizado com o fim de quantificar objetivamente a acurácia dos resultados de segmentação. Especificamente, selecionou-se o Índice Rand Ajustado (*Adjusted Rand Index*, ou ARI) (HUBERT; ARABIE, 1985). Ele mede a similaridade entre a partição proveniente do algoritmo e uma partição de referência, também chamada de *ground truth*. Diferentemente do índice Rand não-ajustado (RAND, 1971), a versão ajustada resulta em valores entre -1 and 1 , e assegura valores próximos a 0 para partições aleatórias ou partições triviais que atribuem todos os elementos a um único grupo, desde que a partição de referência tenha mais de um grupo. Se a partição resultante coincide perfeitamente com a partição de referência, o índice resulta em 1 . Valores negativos, por sua vez, representam alguma anticorrelação entre as duas partições.

Um conjunto de quatro imagens de referência (*ground truth*) foram selecionadas. Com relação à imagem da estrela, dois segmentos são considerados: a forma de estrela e o gradiente

Figura 21 – Visão geral do efeito causado por diferentes ajustes de parâmetros: quantidade de partículas (superpixels) (500 ou 2000) vs. valor do expoente (-16, -12 ou -8) vs. k (2, 3, 4, 5 ou 6). Diferentemente dos demais resultados apresentados neste capítulo, este exemplo específico emprega três dimensões adicionais, além das cinco necessárias, totalizando oito dimensões. Figura presente no trabalho de [Gueleri e Zhao \(2016\)](#).

Figura 22 – Análise de parâmetros para a imagem da estrela. Cada gráfico exibe o valor ARI versus o parâmetro de expoente. Valores ARI iguais a 1 significam que o resultado da segmentação coincide perfeitamente com a imagem de referência (*ground truth*). Valores próximos de 0 equivalem a segmentações aleatórias ou triviais (toda a imagem num único segmento, por exemplo). Os expoentes localizam-se no intervalo entre -16 e -8 , linearmente espaçados a cada $0,2$. Para cada expoente, 30 simulações são realizadas. Cada gráfico sumariza os resultados com os seguintes valores ARI: menor, 1º quartil, mediana, 3º quartil e maior. A linha espessa e contínua representa a mediana. Os gráficos são organizados de acordo com o número de superpixels, o peso atribuído às componentes de cor (*color scale*) e a quantidade k de segmentos. Demais parâmetros: $\alpha_2 = 0,2$ e $\beta = 2$. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Figura 23 – Análise de parâmetros para a imagem do lago. Tanto a metodologia quanto o ajuste de parâmetros são os mesmos da Figura 22. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Figura 24 – Análise de parâmetros para a imagem da torre, particionada em dois segmentos. Tanto a metodologia quanto o ajuste de parâmetros são os mesmos da Figura 22. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Figura 25 – Análise de parâmetros para a imagem da torre, particionada em quatro segmentos. Tanto a metodologia quanto o ajuste de parâmetros são os mesmos da Figura 22. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

de fundo. Para a imagem do lago, três segmentos: o céu, o lago e as construções ao meio. Finalmente, para a imagem da torre, consideram-se duas versões: as construções do plano de frente destacadas do céu de fundo, e uma versão mais refinada na qual o plano de frente é dividido nas três construções que o constituem.

A análise aqui apresentada emprega a seguinte metodologia. O parâmetro de expoente ($\exp$) é o parâmetro que mais afeta a dinâmica que o conjunto de partículas desempenha, consequentemente tendo uma forte influência sobre o resultado final de segmentação. Portanto, cada gráfico é construído variando-se o expoente. Seus valores localizam-se no intervalo entre -16 e -8 , linearmente espaçados a cada $0,2$. Devido à aleatoriedade da atribuição inicial da velocidade das partículas, com descrito na Seção 4.1, diferentes simulações podem conduzir a resultados também diferentes. Portanto, para que se tenha resultados representativos, 30 simulações são realizadas para cada valor de expoente. Os resultados são sumarizados com os seguintes valores ARI: menor, 1º quartil, mediana, 3º quartil e maior. A linha espessa e contínua em cada gráfico representa a mediana.

Os gráficos também são organizados de acordo com o número de partículas (superpixels), o peso atribuído às componentes de cor (*color scale*) e a quantidade k de segmentos. Imediatamente antes da dinâmica começar, como descrito na Seção 4.1, a maneira como se dispõe o conjunto de partículas atribui o mesmo grau de importância às componentes de cor e às coordenadas da imagem. Assim, é possível também mudar esse equilíbrio, a fim de verificar o quanto isso afeta o processo de segmentação. É o que se faz quando se muda a escala de cor. Já o número k é o número de segmentos que se deseja obter logo após a dinâmica ser interrompida.

Procedamos agora para o entendimento desses resultados. Primeiramente, é evidente que os melhores resultados alcançáveis são justamente aqueles em que o número k de segmentos coincide com o número de segmentos da partição de referência. No final das contas, esse é o único caminho para se atingir o valor ARI máximo. Entretanto, dependendo da complexidade da imagem, melhores resultados costumam ser alcançados quando solicita-se uma quantidade diferente de segmentos, geralmente uma quantidade menor. Consideremos a imagem da torre como exemplo, especificamente a versão refinada que contém quatro segmentos de referência. De fato, os maiores valores ARI são encontrados tomando-se quatro segmentos. Mas quando se olha para os resultados em geral, especialmente nos valores medianos, os melhores resultados são encontrados quando solicitam-se apenas dois segmentos, ou três em alguns casos. O motivo é que é muito fácil isolar o céu das construções à frente. Isso não resulta no índice máximo, evidentemente, mas ainda atingem-se valores relativamente altos, em torno de $0,85$. Isso está em conformidade com a ideia de que mesmo não capturando todos os detalhes, capturar a geometria geral da cena pode ser considerado bom o suficiente. Por outro lado, ao tentarmos refinar nossa segmentação solicitando-se um valor maior de k e quebrarmos algum segmento num local inapropriado, seremos severamente penalizados com índices muito baixos. Um exemplo de um refinamento bom e outro ruim, referente à partição de quatro segmentos, é mostrado na

Figura 26 – Além de isolar o céu dos elementos de frente, este exemplo mostra um bom refinamento (em cima) e um refinamento errado que quebra o céu em dois segmentos (embaixo). Este último é responsável por baixos valores ARI. Configuração de parâmetros para ambos os resultados: 1000 superpixels, escala de cor = 1, $\exp = -12$, $\alpha_2 = 0,2$ e $\beta = 2$. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Figura 26.

Aumentando-se o número de superpixels, normalmente consegue-se um pequeno ganho de qualidade. Isso decorre de capturar melhor os pequenos detalhes. Isso também empurra a faixa de bons resultados para menores valores de expoente. No entanto, como discutido na Seção 4.1, menores expoentes dão mais tempo para que o conjunto de velocidades se acomode, consequentemente demandando mais passos para que o sistema atinja seu estado final. Então tem-se uma relação de custo/benefício aqui, um equilíbrio entre acurácia versus tempo de simulação. Além disso, processar mais partículas já demanda por si só mais recursos para calcular cada passo do sistema dinâmico.

Com relação ao peso dado às componentes de cor, seu benefício depende da natureza da imagem. No caso da imagem do lago, por exemplo, é frequentemente vantajoso aumentar o peso atribuído às cores. O lago propriamente dito e o céu têm cores marcantes. Contudo, a atribuição de muita importância às cores tende a romper elementos heterogêneos da imagem, em nosso exemplo, a região entre o lago e o céu. Uma ilustração desse efeito é exibida na Figura 27, algo

Figura 27 – Exemplo de como uma região de cor heterogênea está sujeita à fragmentação. Todos os resultados pertencem à mesma simulação, mas solicitando-se 3, 4 e 7 segmentos. Ajuste de parâmetros: 1000 superpixels, escala de cor = 2, $\exp = -12$, $\alpha_2 = 0,2$ e $\beta = 2$. Trata-se de um dos piores resultados que aparecem na Figura 23, considerando o valor ARI para três segmentos. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

que ocorre mesmo para uma escala de cor moderada ($= 2$) em alguns casos. Nesse exemplo, quando solicitam-se 3 ou 4 segmentos, uma região mais escura e sombreada é desacoplada das demais construções. Para $k = 3$, o resultado é muito ruim. Ainda assim, os resultados são consistentes para maior número de segmentos. Esse efeito pode ser entendido como diferentes interpretações da mesma cena e está em conformidade com o padrão formado pelo conjunto de partículas, que claramente exhibe mais de três grupos. A imagem da torre com 4 segmentos, por outro lado, não revela nenhum ganho ao se aumentar o peso dado às cores. Isso acontece porque as construções do plano de frente não podem ser refinadas em função da cor, mas somente em termos de sua forma.

Figura 28 – Diferentes interpretações das construções. Notar a pequena fachada embaixo do corpo da torre, que pode pertencer tanto à torre que está ao fundo quanto à construção arqueada que está à frente. Ajuste de parâmetros para ambos os resultados: 1000 superpixels, escala de cor = 1, $\exp = -13$, $\alpha_2 = 0,2$ and $\beta = 2$. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Passemos agora à variabilidade de resultados que se obtém para um mesmo ajuste de parâmetros, que também depende das propriedades da imagem. Considerando a imagem da estrela, há um contraste muito grande entre valores ARI muito altos e muito baixos. Índices muito baixos ocorrem quando se erra completamente a forma de estrela, assim dividindo a imagem arbitrariamente. Ainda assim, os resultados medianos se apresentam bem estáveis, exibindo claramente um extenso platô de altos índices. Acontece algo similar com a imagem da torre com 2 segmentos, que exibe grandes contrastes em alguns casos, mas também exibe um platô de altos índices, até mais alto e extenso que aquele da imagem da estrela. Isso sugere que particionar a imagem da torre em dois segmentos é mais fácil que isolar a estrela. Essa variabilidade de resultados, porém, carrega informações importantes, especialmente em casos onde o contraste entre índices baixos e altos não é tão grande. Ela indica diferentes interpretações de uma mesma imagem, e pode ser explorada como um meio de se expressar incerteza. Um exemplo de diferentes interpretações é exibido na Figura 28.

Por fim, apenas para ilustrar que o uso de mais dimensões reduz a chance de colisão entre diferentes grupos de partículas, consideremos a Figura 29, que exibe uma imagem de

Figura 29 – Duas formas distintas contra um plano de fundo amarelo. Tamanho da imagem: 1100x1000. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Figura 30 – Exemplo de colisão entre diferentes partes da imagem. Cinco dimensões são utilizadas. Duas formas distintas colidem entre si e fundem-se num único segmento. Parâmetros: 1000 superpixels, escala de cor = 1, $\exp = -12$, $\alpha_2 = 0,2$ and $\beta = 2$. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Figura 31 – Exemplo de evitar colisão com o uso de mais dimensões. Neste exemplo específico, todos os vetores-velocidade foram inicializados com exatamente os mesmos valores daqueles da Figura 30, exceto pela dimensão adicional aqui utilizada, que acaba influenciando a normalização dos vetores e consequentemente afeta a dinâmica das partículas. Com o uso de uma dimensão adicional, reduz-se a probabilidade de colisão e aquelas duas formas distintas agora tomam direções diferentes, resultando em dois segmentos também distintos. Parâmetros similares aos da Figura 30: 1000 superpixels, escala de cor = 1, $\exp = -12$, $\alpha_2 = 0,2$ and $\beta = 2$. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

entrada contendo duas formas distintas contra um plano de fundo amarelo. Processos dinâmicos utilizando cinco e seis dimensões são ilustrados nas Figuras 30 and 31, respectivamente. Com o uso de uma dimensão adicional (total de seis dimensões), reduz-se a probabilidade de colisão. Aqueles dois grupos de partículas distintos se sobrepõem nas primeiras duas dimensões, possivelmente se sobrepõem também nas três dimensões seguintes (não mostradas na figura), mas o grau de liberdade adicional promovido pela sexta dimensão permite que os dois grupos passem um pelo outro sem colisão, resultando em dois segmentos de imagem distintos. Vale notar, porém, que foi difícil obter o exemplo de colisão mostrado na Figura 30, indicando que cinco dimensões já oferecem um grau de liberdade suficiente na maioria dos casos.

4.2.3 Resultados Comparativos

Pelas Figuras 32, 33, 34 e 35, são apresentados os resultados de dois algoritmos de segmentação cujo escopo de aplicação é similar ao da técnica dinâmica coletiva proposta neste trabalho. Compare essas figuras com as Figuras 22, 23, 24 and 25, da seção de análise de parâmetros, que contêm resultados detalhados da nossa técnica baseada em *flocking*.

Um dos algoritmos é o Potts Segmentation (STORATH; WEINMANN, 2014; STORATH *et al.*, 2015). Os resultados foram obtidos com o auxílio do software Icy¹, utilizando-se seu *plugin* Potts Segmentation². O algoritmo Potts ainda oferece a possibilidade de fusão ou refinamento das partições obtidas na primeira etapa, mas por tratar-se de um procedimento que envolve interação com o usuário, não foi considerado aqui. Apenas a etapa não-interativa foi realizada. Resultados empregando diferentes valores do parâmetro de escala γ são mostrados nos painéis de comparação.

O outro algoritmo é o chamado Felzenszwalb (FELZENSZWALB; HUTTENLOCHER, 2004). Trata-se de um algoritmo de segmentação baseado em grafo. Os resultados foram obtidos pelo emprego da biblioteca *scikit-image*³, uma coleção de algoritmos para processamento de imagens escritos em python (van der Walt *et al.*, 2014). A versão 0.9.3 foi utilizada. Nas Figuras 32, 33, 34 e 35, mostram-se os valores ARI versus diferentes valores do parâmetro de escala e diferentes valores de σ .

Há ainda outros algoritmos famosos de segmentação de imagens, por exemplo o Watershed Transformation (BEUCHER; MEYER, 1993), mas ele ou requer marcadores iniciais definidos pelo usuário, ou geralmente resulta numa segmentação excessiva (*over-segmentation*) quando automatiza-se o processo escolhendo-se, por exemplo, mínimos locais como marcadores iniciais, o que acaba requerendo algum processo sofisticado de fusão de regiões para reduzir o número de segmentos. Outro método famoso é o Random Walker (GRADY, 2006), mas ele também requer marcadores iniciais definidos pelo usuário. Os algoritmos Felzenszwalb e Potts

¹ <<http://icy.bioimageanalysis.org/>>

² <http://icy.bioimageanalysis.org/plugin/Potts_Segmentation/>

³ <<http://scikit-image.org/>>

Figura 32 – Comparação de acurácia com a imagem da estrela. São apresentados os resultados da aplicação do algoritmo Potts Segmentation (STORATH; WEINMANN, 2014; STORATH *et al.*, 2015) e do algoritmo de segmentação baseado em grafo conhecido como Felzenszwalb (FELZENSZWALB; HUTTENLOCHER, 2004). Para o algoritmo Potts, diferentes valores do parâmetro de escala γ são utilizados. Os demais parâmetros são mantidos em seus valores *default*: 0,001 para o valor inicial de μ , 2 para o valor de passo de μ e isotropismo ativado. Para o algoritmo Felzenszwalb, diferentes valores de seu parâmetro de escala são utilizados também. Além disso, as diferentes linhas do gráfico representam resultados para diferentes valores de σ , partindo de 0,1 (azul) até 1,0 (vermelho), linearmente espaçados a cada 0,1. O menor tamanho de componente é mantido em seu valor *default*: 20. Figura presente no trabalho de Gueleri *et al.* (2018).

Segmentation foram escolhidos porque estão facilmente disponíveis e porque eles compartilham com nossa técnica o mesmo escopo de aplicação, isto é, eles são não-supervisionados — não requerem marcadores iniciais definidos pelo usuário — e geralmente não resultam em segmentação excessiva.

O Felzenszwalb funciona realmente bem na imagem da estrela, enquanto o Potts é incapaz de distinguir a estrela do gradiente de fundo. Na imagem do lago, o Potts é novamente incapaz de atingir bons índices acima de 0,7. O Felzenszwalb atinge índices altos, mas somente dentro de uma faixa bem limitada de parâmetros, enquanto nossa técnica baseada em *flocking* atinge índices altos numa faixa bem mais ampla. Com relação à imagem da torre, de 2 segmentos, o Felzenszwalb é capaz de atingir índices muito altos, mas novamente dentro de uma faixa bem limitada de parâmetros. O Potts é capaz de atingir resultados razoáveis, mas não tão altos. Nossa técnica *flocking*, por outro lado, não somente é capaz de atingir índices muito altos, mas

Figura 33 – Comparação de acurácia com a imagem do lago. Tanto a metodologia quando a configuração de parâmetros são similares aos da Figura 32. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

também o faz numa faixa muito mais ampla de parâmetros. Finalmente, na imagem da torre de 4 segmentos, nem o Felzenszwalb nem o Potts são capazes de revelar os quatro segmentos. Diferentemente da nossa técnica, que é capaz de encontrar os quatro segmentos em algumas simulações, os índices mais altos que o Felzenszwalb e o Potts podem atingir são justamente equivalentes aos nossos quando solicitam-se apenas dois segmentos.

Figura 34 – Comparação de acurácia com a imagem da torre, considerando dois segmentos. Tanto a metodologia quando a configuração de parâmetros são similares aos da Figura 32. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

Figura 35 – Comparação de acurácia com a imagem da torre, considerando quatro segmentos. Tanto a metodologia quando a configuração de parâmetros são similares aos da Figura 32. Figura presente no trabalho de [Gueleri et al. \(2018\)](#).

DETECÇÃO DE COMUNIDADES VIA REMOÇÃO ITERATIVA DE ARESTAS

A presença e o estudo de estrutura de comunidades são comumente encontrados em diversas redes reais, tais como redes sociais, redes de interação de proteínas na biologia e na bioinformática (JONSSON *et al.*, 2006; ZHANG, 2009), estruturas de fluxo óleo-água (GAO *et al.*, 2015), redes de mobilidade humana (THIEMANN *et al.*, 2010), estrutura espacial de movimento urbano em grandes cidades (ZHONG *et al.*, 2014), redes de elites corporativas nos campos da política e da economia (HEEMSKERK; TAKES, 2016), e muito mais. Comunidades, também conhecidas como estruturas modulares, são comumente definidas como sendo grupos de vértices densamente conectados (CLAUSET; NEWMAN; MOORE, 2004; NEWMAN; GIRVAN, 2004; COSTA *et al.*, 2007; BLONDEL *et al.*, 2008; FORTUNATO, 2010; LANCICHINETTI *et al.*, 2010; SILVA; ZHAO, 2012c; QUILES; MACAU; RUBIDO, 2016; SILVA; ZHAO, 2016). Como consequência, conexões entre comunidades diferentes devem ser mais esparsas. As definições podem levemente variar se forem considerados, por exemplo, casos mais gerais de comunidades sobrepostas, nas quais um único elemento pode pertencer a mais de uma comunidade ao mesmo tempo. Por exemplo, esse é o caso de redes sociais, onde uma única pessoa pertence ao seu grupo de família, ao seu grupo de colegas de trabalho etc. No trabalho aqui apresentado, entretanto, considera-se apenas o caso de estrutura de comunidades sem sobreposição.

O problema de detecção de comunidades (não-sobrepostas) está relacionado ao problema de partição de grafo em teoria dos grafos. Encontrar a partição ótima é considerado um problema NP-difícil na maioria dos casos (FORTUNATO, 2010), consequentemente abrindo espaço para ampla pesquisa em meios de se encontrar soluções sub-ótimas em tempo admissível. Diferentes abordagens para detecção de comunidades já foram desenvolvidas, incluindo propriedades espectrais em matrizes de grafos (DONETTI; MUÑOZ, 2004; CAPOCCI *et al.*, 2005; YANG; LIU, 2008; MITROVIĆ; TADIĆ, 2009), sincronização de osciladores acoplados (ARENAS;

DÍAZ-GUILERA; PÉREZ-VICENTE, 2006), modelo de Potts (REICHARDT; BORNHOLDT, 2004), caminhada e competição de partículas em redes (QUILES *et al.*, 2008; BREVE; ZHAO; QUILES, 2009; SILVA; ZHAO, 2012c; SILVA; ZHAO, 2016; VERRI; URIO; ZHAO, 2016), e dinâmica de partículas com forças de atração e repulsão (QUILES; MACAU; RUBIDO, 2016).

Uma abordagem bastante conhecida para detecção de comunidades baseia-se no conceito de *betweenness* de aresta, que mensura a relevância de uma aresta em função da quantidade de caminhos-mínimos que incluem tal aresta (FREEMAN, 1977; NEWMAN; GIRVAN, 2004; COSTA *et al.*, 2007; FORTUNATO, 2010). O algoritmo foi desenvolvido por Newman e Girvan (2004) e pressupõe que a aresta de maior *betweenness* é a que mais provavelmente conecta comunidades distintas. O algoritmo então funciona calculando o grau de *betweenness* de todas as arestas, removendo a de maior *betweenness*, calculando novamente sobre o novo conjunto de arestas, e assim sucessivamente. A principal deficiência dessa abordagem é, no entanto, sua complexidade de tempo. Numa rede contendo m arestas e n vértices, o algoritmo demanda $O(m^2n)$, ou $O(n^3)$ em redes esparsas, onde $m \sim n$ (CLAUSET; NEWMAN; MOORE, 2004; NEWMAN; GIRVAN, 2004; LANCICHINETTI; FORTUNATO, 2009; FORTUNATO, 2010).

Newman e Girvan (2004) também propuseram uma métrica denominada *modularidade*, cujo propósito é quantificar a tendência da rede em exibir estrutura de comunidades (CLAUSET; NEWMAN; MOORE, 2004; NEWMAN; GIRVAN, 2004; COSTA *et al.*, 2007; BLONDEL *et al.*, 2008; FORTUNATO, 2010; LANCICHINETTI; FORTUNATO, 2011). Essa métrica não assume nenhum conhecimento *a priori* da rede, por exemplo, rótulos de vértices. Embora limitações em suas propriedades tenham sido apontadas (LANCICHINETTI; FORTUNATO, 2011), essa métrica é empregada em diversas abordagens de detecção de comunidades. Um dos algoritmos utilizados neste trabalho de doutorado, tanto para comparação como para etapa complementar, é chamado “Cluster Fast Greed”, ou simplesmente CFG, e é baseado em uma otimização gulosa da modularidade (CLAUSET; NEWMAN; MOORE, 2004; LANCICHINETTI; FORTUNATO, 2009). Ele é bem rápido, tendo custo $O(md \log n)$, onde d é a profundidade do “dendrograma” que descreve a estrutura hierárquica de comunidades retornada pelo algoritmo. Em situações onde $m \sim n$ e $d \sim \log n$, ele roda em tempo quasilinear: $O(n \log^2 n)$. Outro algoritmo aqui empregado é o chamado “Louvain”, que também é baseado na otimização da modularidade (BLONDEL *et al.*, 2008; LANCICHINETTI; FORTUNATO, 2009). Seus autores defendem seu custo computacional, o que o torna aplicável em redes muito grandes contendo milhões de vértices.

A técnica de detecção de comunidades apresentada neste capítulo é dividida em duas etapas alternadas. A primeira é um processo dinâmico coletivo que toma como inspiração a formação *flocking* que pode ser vista na natureza, por exemplo, em grandes grupos de pássaros ou de peixes (VICSEK *et al.*, 1995; MISHRA *et al.*, 2012; SUMINO *et al.*, 2012; TONER, 2012a; VICSEK; ZAFEIRIS, 2012; NAGAI *et al.*, 2015), e que vem sendo estudada ao longo deste trabalho de doutorado. Em nosso modelo, faz-se cada vértice ser uma partícula que se alinha

num sistema dinâmico coletivo. Portanto, cada vértice carrega um vetor-velocidade, que começa apontando para uma direção aleatória e, conforme o processo avança, progressivamente vai tomando a mesma direção de seus vizinhos. Na verdade, diferentemente das técnicas apresentadas nos demais capítulos, os vértices não se movimentam neste modelo, portanto o termo “velocidade” é apenas uma analogia à direção de movimentação em sistemas *flocking*, cujas partículas tentam se alinhar com seus vizinhos, ou seja, tentam tomar a mesma direção de movimento.

O processo dinâmico é interrompido após certo tempo, então entra a segunda etapa. Nela, mede-se o (des)alinhamento de cada par de vértices diretamente conectados, ou seja, par cujos vértices interagem diretamente um com o outro. A aresta responsável pelo par cujos vértices se tornaram menos alinhados é supostamente uma aresta que conecta comunidades distintas, portanto ela é removida e a primeira etapa entra em ação novamente (sem a aresta removida). Ao invés de apenas uma aresta por vez, é possível também economizar tempo removendo-se uma fração maior dos vértices responsáveis pelos mais altos desalinhamentos. Após um número suficiente de ciclos de remoção, espera-se que a maioria das arestas inter-comunidades já tenham sido removidas, tornando assim mais fácil o processo final de partição da rede. Em alguns casos, a rede já se torna particionada em componentes (sub-redes) completamente desconectadas umas das outras. Na maioria das situações, entretanto, ainda é necessário o emprego de algum algoritmo auxiliar para obter a partição final. Aplicar tal algoritmo após a devida remoção das arestas é mais efetivo que aplicá-lo na rede original, evidentemente, caso contrário não faria sentido a técnica de remoção de arestas aqui discutida. Como mencionado há pouco, esta pesquisa faz intenso uso de dois algoritmos tradicionais de detecção de comunidades: CFG e Louvain.

Os resultados deste capítulo deram origem ao seguinte trabalho, ainda em fase de submissão:

Roberto Alves Gueleri, Filipe Alves Neto Verri, Qiusheng Zheng, Junbao Zhang, and Liang Zhao. **Network community detection via iterative edge removal in a flocking-like system.** Em fase de submissão.

O restante deste capítulo está organizado da seguinte forma. Na Seção 5.1, apresentam-se o modelo dinâmico e o processo de remoção de arestas em detalhes. Um conjunto abrangente de resultados experimentais é apresentado na Seção 5.2. Nessa seção, estudam-se como o desalinhamento dos vértices evolui com o passar do tempo (Seção 5.2.1), a influência de diferentes configurações de parâmetros (tanto parâmetros de algoritmo quanto parâmetros de redes geradas artificialmente) (Seções 5.2.3, 5.2.5, and 5.2.6), assim como estuda-se também a complexidade computacional do processo iterativo de remoção de arestas como um todo (Seção 5.2.4).

5.1 Descrição do Modelo

Apresenta-se aqui a abordagem iterativa de remoção de arestas em detalhes. Como dito anteriormente, o processo é dividido em duas etapas alternadas. O modelo dinâmico de alinhamento de partículas é apresentado logo a seguir, na Seção 5.1.1. Já a etapa de remoção de arestas propriamente dita é apresentada na Seção 5.1.2.

5.1.1 O Modelo Dinâmico de Alinhamento de Partículas

Dada uma rede não-direcionada, livre de autoconexões (conexões de um vértice para ele mesmo) e arestas múltiplas (mais de uma conexão entre um mesmo par de vértices), faz-se cada vértice ser uma partícula que se alinha em um sistema dinâmico coletivo. Assim, cada vértice carrega um vetor-velocidade $\mathbf{v}_i(t)$, que aponta para alguma direção num espaço multidimensional. Na verdade, os vértices não se movimentam, portanto o termo “velocidade” é apenas uma analogia à direção de movimentação em sistemas *flocking*, como já mencionado anteriormente.

As arestas definem a vizinhança de cada partícula i , isto é, as partículas com as quais i interage diretamente. A vizinhança não muda ao longo do processo dinâmico, portanto a rede de interação é sempre a mesma.

Inicialmente, a cada partícula i é associada uma velocidade inicial aleatória $\mathbf{v}_i(0)$, que é um vetor unitário apontando para alguma direção aleatória. Um espaço tridimensional é utilizado para todos os experimentos apresentados neste trabalho. Contudo, o impacto que diferentes dimensionalidades tem no resultado final é algo ainda a ser investigado.

O processo dinâmico é regido pela seguinte expressão:

$$\mathbf{v}_i(t+1) = \hat{\mathbf{v}}_i(t) + \alpha \cdot \frac{1}{k_i} \cdot \sum_j (\hat{\mathbf{v}}_j(t) - \hat{\mathbf{v}}_i(t)) \quad (5.1)$$

O termo $\hat{\mathbf{v}}_i(t)$ denota um vetor unitário, isto é, $\hat{\mathbf{v}}_i(t) = \mathbf{v}_i(t) / \|\mathbf{v}_i(t)\|$. O parâmetro t é o índice de iteração (tempo), que começa valendo zero. O parâmetro α define o quão rápido atualizam-se as velocidades. A somatória ocorre sobre todos os vizinhos j . O valor k_i é o grau do vértice i , ou seja, o número de vizinhos que ele tem. Portanto, $1/k_i$ é um termo de normalização, evitando que partículas de alto grau se alinhem mais rápido que as outras de menor vizinhança.

Embora as partículas não se movam — não há nem posição espacial a elas definida —, a utilização do vetor unitário $\hat{\mathbf{v}}_i(t)$ impõe algo análogo ao “movimento de velocidade constante” que cada partícula desempenharia. Além disso, todas as partículas se locomoveriam com a mesma velocidade. Movimento de velocidade constante é uma característica comumente modelada em estudos de partículas autopropulsionadas (VICSEK *et al.*, 1995; SUMINO *et al.*, 2012; TONER, 2012a; VICSEK; ZAFEIRIS, 2012; NAGAI *et al.*, 2015). Há estudos, porém, que exploram sistemas autopropulsionados com velocidade variável (MISHRA *et al.*, 2012), mas tal ramo

de pesquisa está fora do escopo do nosso trabalho. Mais ainda, a normalização imposta aos vetores-velocidade faz do sistema um sistema não-linear. Isso é também responsável pela quebra de simetria rotacional que faz com que todo o conjunto de partículas concorde em tomar uma mesma direção em suas velocidades. Caso contrário, sem a imposição de tal normalização, todos os vetores-velocidade se extinguiriam em zero — ou perto de zero — devido à sua distribuição inicial aleatória, fazendo com que vetores opostos neutralizem uns aos outros.

Uma ilustração do processo dinâmico é mostrada na Figura 36. Embora o layout mostrado isole as quatro comunidades, é importante lembrar que o sistema dinâmico apresentado aqui não “conhece” previamente nenhuma comunidade. Descobrir a estrutura de comunidades subjacente é, acima de tudo, o principal propósito desse sistema. Além disso, o objeto de entrada é formado somente pelo conjunto de vértices e arestas, ou seja, a topologia da rede, portanto nenhuma informação de localização espacial dos vértices é fornecida.

Mesmo vendo que todos os vetores-velocidade parecem estar completamente alinhados após 500 passos na Figura 36, pelas seções seguintes será mostrado que, numa escala “microscópica”, os vetores ainda apontam para diferentes direções. Mais ainda, sua disposição permite identificar as arestas que provavelmente conectam comunidades distintas, tornando possível, pois, isolar tais comunidades.

5.1.2 O Processo Iterativo de Remoção de Arestas

Define-se *entropia* como sendo o nível de “desordem” em termos do desalinhamento entre os vetores-velocidade. Cada par cujos vértices interagem diretamente contribui para a entropia total do sistema. Assim, a entropia total é a composição de cada entropia individual $H_e(t)$ definida para cada aresta e :

$$H_e(t) = \sum_{d=1}^D |\mathbf{v}_{i,d}(t) - \mathbf{v}_{j,d}(t)| \quad (5.2)$$

D é o número de dimensões, enquanto $\mathbf{v}_{i,d}(t)$ e $\mathbf{v}_{j,d}(t)$ são os vetores-velocidade associados aos vértices conectados pela aresta e , projetados na dimensão d . Em outras palavras, $H_e(t)$ é simplesmente a distância bloco-cidade, ou distância L_1 , entre os vetores $\mathbf{v}_i(t)$ e $\mathbf{v}_j(t)$. O conceito de entropia discutido aqui apenas captura a ideia de desordem mensurada em termos de desalinhamento entre vetores, e não deve ser confundido com outras definições rigorosas de campos como termodinâmica, mecânica estatística ou teoria da informação. Mais ainda, a menos que seja dito explicitamente, por “entropia” entende-se entropia de aresta e não a entropia total.

Por meio dos experimentos apresentados na próxima seção, é possível ver que as entropias de arestas que conectam comunidades distintas se tornam geralmente maiores que aquelas de arestas dentro de uma mesma comunidade. Isso forma a base de nossa abordagem iterativa de remoção de arestas: a remoção das arestas de maior entropia provavelmente elimina arestas inter-comunidades, consequentemente tornando a distinção entre diferentes comunidades cada

Figura 36 – Ilustração do processo de alinhamento. A primeira imagem é a rede de entrada: 4 comunidades, cada uma formada por 200 vértices, grau médio $\langle k \rangle \approx 10$ e probabilidade de conexão dentro da mesma comunidade $p_{\text{in}} = 0,60$ (consequentemente, $p_{\text{out}} = 1 - p_{\text{in}} = 0,40$). As imagens subsequentes mostram o estado inicial em $t = 0$, com os vetores-velocidade apontando para direções aleatórias, e estados subsequentes em $t = 100$ e $t = 500$. Por uma questão de clareza, as arestas da rede são omitidas dessas últimas imagens. As velocidades têm três graus de liberdade — três dimensões —, mas somente a projeção em duas dimensões é mostrada. Parâmetro $\alpha = 0,1$. Embora todos os vetores-velocidade pareçam estar completamente alinhados após 500 passos, numa escala “microscópica” eles ainda apontam para diferentes direções, o que permite identificar as arestas que provavelmente conectam comunidades distintas.

vez mais clara com o passar dos ciclos de remoção. Assim, o processo de remoção de arestas como um todo consiste nos seguintes passos:

1. Após associar vetores-velocidade inicialmente aleatórios para todos os vértices, executa-se o modelo dinâmico de alinhamento de partículas, como descrito na Seção 5.1.1, até um certo número desejado de passos.
2. Assim que a dinâmica é interrompida, coletam-se as entropias de todas as arestas. É possível executar o modelo dinâmico (Passo 1) múltiplas vezes, usando diferentes atribuições aleatórias do conjunto de velocidades iniciais. Nesse caso, a entropia final de cada aresta pode ser simplesmente a soma das entropias individuais coletadas ao final de cada simulação.
3. Removem-se as arestas de maior entropia, então retorna-se ao Passo 1 e executa-se o modelo dinâmico novamente, mas dessa vez usando a nova rede sem as arestas removidas.

Os Passos 1, 2 e 3, juntos, formam o que chamamos de “ciclo” (*round*). Na Seção 5.2.3, estudam-se os efeitos de se executar o modelo dinâmico múltiplas vezes por ciclo. Estudam-se também os efeitos de se remover diferentes quantidades de arestas por ciclo.

5.2 Resultados Experimentais

Apresenta-se agora um amplo conjunto de resultados experimentais que foram conduzidos sobre diferentes classes de redes geradas por computador. Como mencionado anteriormente, os experimentos empregam dois algoritmos tradicionais e bem conhecidos, tanto para comparação quanto para etapa complementar: “Cluster Fast Greed” (CFG) (CLAUSET; NEWMAN; MOORE, 2004) e “Louvain” (BLONDEL *et al.*, 2008). Sempre que aplicado na rede original, sem qualquer modificação, eles nos fornecem resultados comparativos. Além disso, podemos aproveitar esses algoritmos — e outros da literatura também — empregando-os como etapa complementar, isto é, após uma fração de arestas ser eliminada pelo processo de remoção de arestas aqui apresentado, é possível aplicar algum algoritmo alternativo de detecção de comunidades para se obter a partição final da rede. Espera-se que a tarefa de obtenção de comunidades se torne mais fácil após as arestas serem apropriadamente removidas. Lembrar que procura-se remover arestas inter-comunidades. Alternativamente, o processo de remoção de arestas também pode por si só resultar nas comunidades. É necessário apenas esperar um pouco mais até que cada comunidade esteja completamente desconectada das outras. Desse modo, não é nem necessário o auxílio de um algoritmo externo de detecção de comunidades. Resultados de ambas as abordagens são apresentados nesta seção.

5.2.1 Evolução da Entropia ao longo do Tempo

Na seção anterior, foi dito que a entropia de arestas que conectam comunidades distintas geralmente se torna maior que aquela de arestas dentro de uma mesma comunidade. Apresenta-se agora, na Figura 37, a maneira como a entropia das arestas evolui com o passar do tempo. A fim de se reduzir a dependência sobre a condição inicial — atribuição aleatória de velocidades —, a população dos valores de entropia é obtida a partir de 10 simulações independentes. A rede de entrada, no entanto, é exatamente a mesma para todas essas simulações. É possível ver que as entropias decaem exponencialmente até atingirem uma configuração aparentemente estável e constante. No exemplo mostrado na Figura 37, após cada ciclo, aproximadamente 1% das arestas são removidas. Precisamente, 40 arestas são removidas por ciclo. A rede tem 800 vértices e $\langle k \rangle \approx 10$, resultando num total de aproximadamente 4000 arestas. Observa-se que, de fato, arestas inter-comunidades tomam valores de entropia normalmente maiores que arestas intra-comunidades.

Nos gráficos superiores da Figura 37, a linha horizontal pontilhada denota o *machine epsilon* de números em ponto flutuante de dupla precisão (64-bits), cujo valor é aproximadamente $2,22 \times 10^{-16}$. Diferentes definições são encontradas na literatura, mas Higham (2002) (página 37) define *machine epsilon* como sendo a distância entre 1,0 e o número em ponto flutuante imediatamente maior. Ele está relacionado ao erro de arredondamento que surge ao se operar um par de números de escalas muito diferentes (GOLDBERG, 1991; HIGHAM, 2002). Em nosso caso, a entropia, que mede a diferença entre um par de vetores, decai exponencialmente ao passo que α é constante ao longo de todo o processo. É evidente que a diferença em termos de escala entre a entropia e α acabará sendo suficientemente grande para impedir que a entropia continue diminuindo gradativamente, tornando-a ou zero ou algum valor próximo de *machine epsilon*. No caso de uma máquina de precisão ilimitada, a entropia diminuiria gradativamente e indefinidamente, cada vez mais próxima de zero, sem ficar presa em torno de *machine epsilon* como exibido na Figura 37. Investigações preliminares revelaram que, de fato, não há mudança em nenhum vetor-velocidade assim que o sistema atinge esse estado aparentemente estável. A partir da Figura 37, é possível observar também que quando o sistema entra nessa configuração estável e constante ao redor de *machine epsilon*, os gráficos *inter-/intra-community entropy ratio* exibem uma transição abrupta em seu regime. Ainda assim, tanto antes quanto depois dessa transição, arestas inter-comunidades mantêm maiores entropias. Entretanto, como é discutido mais à frente na Seção 5.2.5, a interrupção do processo de alinhamento antes que o sistema atinja tal configuração estável pode levar a resultados muito insatisfatórios, mesmo sabendo que as entropias inter-comunidades já são maiores que as intra-comunidades.

5.2.2 Visão Geral dos Resultados de Detecção de Comunidades

Alguns resultados provenientes da aplicação do processo iterativo de remoção de arestas são apresentados nas Figuras 38 e 39. Todas as redes de entrada consideradas neste trabalho são

Figura 37 – Evolução da entropia da rede apresentada na Figura 36: 4 comunidades, cada uma formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$. A população de valores de entropia é obtida de 10 simulações independentes. Parâmetro $\alpha = 0,1$. Gráficos superiores: a evolução da entropia separada entre arestas intra e inter-comunidades. São exibidos os valores: menor, 1º quartil, mediana, 3º quartil e maior. O eixo-y está em escala logarítmica, portanto valores exatamente iguais a zero são omitidos. Isso acontece onde as linhas de menor entropia são interrompidas. Gráficos intermediários: no caso da mediana, por exemplo, mostra-se a razão $\frac{\text{mediana inter-comunidade}}{\text{mediana intra-comunidade}}$. Similarmente, mostram-se as razões para o 3º quartil e para o maior valor. Claramente, valores acima de 1 indicam que arestas conectando diferentes comunidades tomam valores maiores de entropia, ou seja, os vértices que elas conectam se tornam menos alinhados. Uma vez que somente as arestas de maior entropia serão removidas, não há interesse em mostrar os resultados de menor entropia. Gráficos inferiores: a distribuição de entropia em $t = 5000$. Cada linha é a interpolação de um histograma de 200 classes (200 bins). Gráficos à esquerda: simulações no primeiro ciclo, ou seja, antes que qualquer aresta tenha sido removida. Gráficos à direita: simulações no ciclo número 10, ou seja, após 9 ciclos de remoção de arestas.

Figura 38 – Visão geral dos resultados de detecção de comunidades. A rede de entrada é exatamente a mesma mostrada na Figura 36: 4 comunidades, cada uma formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$. Exibem-se os resultados da aplicação dos algoritmos CFG e Louvain na rede original. Comunidades são representadas por diferentes cores. Na última figura, mostra-se o resultado da aplicação da técnica iterativa de remoção de arestas. Nesse caso, as comunidades são simplesmente componentes conexos distintos. Em cada ciclo, 10 simulações independentes são executadas e 40 arestas ($\approx 1\%$) são removidas. Parâmetro $\alpha = 0,1$.

Figura 39 – Visão geral dos resultados de detecção de comunidades numa situação um pouco mais fácil: $p_{in} = 0,66$. Todos os demais parâmetros e o significado das figuras são similares aos da Figura 38.

construídas empregando-se a seguinte metodologia:

1. Dado um grau médio desejado $\langle k \rangle_{\text{des}}$, cada vértice i escolhe aleatoriamente $\langle k \rangle_{\text{des}}/2$ vértices j para se conectar. Para cada j a ser escolhido, j é tomado da mesma comunidade de i com probabilidade p_{in} , ou similarmente tomado de outra comunidade com probabilidade $p_{\text{out}} = 1 - p_{\text{in}}$. A escolha de um mesmo vértice j duas ou mais vezes é permitida, como também é permitido que i se conecte com ele mesmo. Além disso, j sendo escolhido por i não impede que i também seja escolhido por j .
2. Após estabelecer todas as conexões, a rede é então simplificada, no sentido de que autoconexões são removidas e múltiplas arestas que conectam um mesmo par de vértices tornam-se uma única aresta não-direcionada. Como resultado, o grau médio definitivo $\langle k \rangle$ pode ser reduzido, mas para redes grandes, ele ainda se mantém bem próximo do grau desejado $\langle k \rangle_{\text{des}}$.

Para fins de comparação, os resultados da aplicação dos algoritmos Cluster Fast Greedy (CFG) (CLAUSET; NEWMAN; MOORE, 2004) e Louvain (BLONDEL *et al.*, 2008) sobre a rede original também são mostrados nas Figuras 38 e 39. Uma vez que ambos os métodos funcionam por meio da otimização da medida de modularidade, eles simplesmente retornam a partição de melhor modularidade encontrada. Observando as cores dos vértices, que denotam o conjunto de comunidades encontradas, pode-se dizer que os resultados da técnica iterativa de remoção de arestas são melhores que aqueles dos algoritmos CFG ou Louvain. Na próxima seção, contudo, são apresentados mais resultados em termos de um índice objetivo: o índice Rand ajustado.

Nos experimentos aqui apresentados, os identificadores (IDs) dos vértices são organizados de acordo com suas comunidades, isto é, todos os vértices da primeira comunidade aparecem primeiro, seguidos por todos os vértices da segunda comunidade, e assim por diante. Essa ordem não tem nenhuma influência no processo dinâmico apresentado neste trabalho. Entretanto, é importante notar que o algoritmo Louvain pode ser influenciado pela ordem dos vértices (BLONDEL *et al.*, 2008). Ainda assim, de acordo com seus autores, a sequência dos vértices não parece influenciar significativamente a modularidade obtida (BLONDEL *et al.*, 2008). E uma vez que o algoritmo Louvain é baseado na otimização da modularidade, a ordem dos vértices não deve influenciar significativamente a partição resultante.

5.2.3 Análise de Parâmetros

Resultados obtidos com o emprego de diferentes configurações de parâmetros são apresentados nesta seção. Um índice de avaliação é utilizado com o fim de quantificar objetivamente a acurácia do conjunto de comunidades obtido. Especificamente, selecionou-se o Índice Rand Ajustado (*Adjusted Rand Index*, ou ARI) (HUBERT; ARABIE, 1985), o mesmo apresentado e utilizado no Capítulo 4 para medir a qualidade dos segmentos de imagem.

Aplicações reais, contudo, não oferecem tal partição de referência, então emprega-se aqui também a medida de *modularidade*, que é uma métrica proposta por Newman e Girvan (2004), cujo propósito é quantificar a tendência da rede em exibir estrutura de comunidades (CLAUSET; NEWMAN; MOORE, 2004; NEWMAN; GIRVAN, 2004; COSTA *et al.*, 2007; BLONDEL *et al.*, 2008; FORTUNATO, 2010; LANCICHINETTI; FORTUNATO, 2011). Diferentemente do ARI, a medida de modularidade não assume nenhum conhecimento *a priori* da rede, por exemplo, rótulos de vértices. Colocando os valores ARI e de modularidade lado-a-lado, pode-se então procurar por alguma relação entre eles, ou seja, procurar por alguma relação entre uma informação disponível numa aplicação real (modularidade) e uma medida robusta baseada em partições de referência (ARI).

Consideremos primeiramente a mesma rede apresentada na Figura 36, cujo $p_{in} = 0,60$. Valores ARI e de modularidade com o passar dos ciclos são apresentados na Figura 40, organizados conforme o número de simulações independentes por ciclo e conforme a fração de arestas removidas a cada ciclo. Uma melhoria nítida pode ser verificada quando os algoritmos CFG ou Louvain são aplicados na rede previamente transformada pelo processo de remoção de arestas. Mesmo considerando-se as componentes conexas, sem o auxílio de qualquer algoritmo externo, pode-se ver resultados claramente melhores que aqueles provenientes da aplicação dos algoritmos CFG e Louvain na rede original. No entanto, a aplicação do algoritmo CFG na rede transformada parece ser a melhor abordagem na maioria dos casos.

Como pode-se observar, o valor de modularidade resultante tanto do algoritmo CFG quanto do Louvain não fornece muita informação. Ele permanece praticamente constante ao longo dos diferentes ciclos. Entretanto, a modularidade mostra variações nítidas quando proveniente das componentes conexas da rede, que são obtidas pelo próprio processo de remoção de arestas. Mais ainda, uma relação desejável pode ser notada aqui: a maior modularidade das componentes conexas coincide com valores ARI relativamente altos. De modo mais preciso, por maior modularidade entende-se a *primeira maior modularidade* (maior local, não necessariamente global), ou seja, o valor da modularidade imediatamente antes que ela comece a decrescer, como é o caso de exemplos tratados mais à frente neste capítulo. Esse resultado é útil para se estabelecer o ciclo ideal após o qual uma boa partição pode ser obtida: o ciclo imediatamente anterior à diminuição do valor de modularidade, tomando-se o estado da rede nesse ciclo como sendo a rede de entrada para algum algoritmo de detecção de comunidades, por exemplo o CFG ou o Louvain, ou simplesmente tomando-se as componentes conexas como sendo o conjunto final de comunidades. Embora os resultados da Figura 40 sejam originários de uma única instância de rede, o mesmo padrão geral está presente em outras instâncias, mesmo utilizando-se diferentes parâmetros de rede, como pode ser visto nos demais resultados apresentados neste capítulo.

Com relação ao número de simulações independentes por ciclo e a fração de arestas removidas a cada ciclo, não são encontradas grandes diferenças entre diferentes configurações, embora a remoção de uma menor quantidade de arestas por ciclo pareça ser ligeiramente melhor.

Figura 40 – Valores ARI e de modularidade ao longo dos diferentes ciclos (*rounds*). A rede de entrada é exatamente a mesma mostrada na Figura 36: 4 comunidades, cada uma formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$. Os gráficos são organizados conforme o número de simulações independentes por ciclo (1 ou 10) e conforme a fração de arestas removidas a cada ciclo (uma única aresta, 10 arestas ($\approx 0,25\%$) ou 40 arestas ($\approx 1\%$)). As linhas finas e horizontais nos gráficos ARI são linhas de referência que correspondem à aplicação dos algoritmos CFG e Louvain na rede original, ou seja, antes que qualquer aresta seja removida. Elas nos oferecem resultados comparativos. As demais linhas apresentam os resultados ao longo dos ciclos de remoção de arestas. Além dos algoritmos CFG e Louvain, apresentam-se também os resultados “Component” em que consideram-se as componentes conexas da rede, que são diretamente obtidas da técnica de remoção de arestas. Em todas as simulações, parâmetro $\alpha = 0,1$.

Figura 41 – Comparação entre remover 10 arestas ($\approx 0,25\%$) e 40 arestas ($\approx 1\%$) por ciclo numa população de 50 redes diferentes. Cada rede tem 4 comunidades, cada comunidade formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$. As caixas em vermelho são os resultados da aplicação dos algoritmos CFG e Louvain na rede original, ou seja, antes que qualquer aresta tenha sido removida. Eles fornecem resultados comparativos. As demais caixas (em azul) são obtidas pela aplicação do algoritmo CFG após um certo número de ciclos de remoção de arestas. Notar que remover 0,25% ou 1%, e executar respectivamente 120 ou 30 ciclos, ambos os casos terminam removendo o mesmo número total de arestas. Uma única simulação por ciclo é realizada, e o parâmetro $\alpha = 0,1$ para toda simulação.

Por outro lado, remover uma maior fração de arestas é menos custoso, pois demanda menos ciclos para completar. Na Figura 41, é apresentada uma comparação entre remover aproximadamente 0,25% e 1% de arestas por ciclo numa população de 50 redes diferentes. Realmente, a remoção de uma menor fração de arestas parece ser ligeiramente melhor, especialmente considerando os valores medianos. Além disso, a variabilidade de resultados é reduzida quando se remove menos arestas por ciclo. Em todo caso, porém, o processo de remoção de arestas conduz a resultados consideravelmente melhores quando comparado aos resultados da aplicação dos algoritmos CFG ou Louvain na rede original, sem modificação.

Resultados para redes de diferentes probabilidades de conexão intra-comunidade (p_{in}) são apresentados na Figura 42. A fim de garantir resultados mais consistentes, não sendo afetados pela configuração aleatória inicial, cada gráfico é obtido rodando-se um conjunto de 10 simulações independentes por ciclo, mesmo sabendo que os resultados da Figura 40 sugerem que não faz muita diferença se comparado aos resultados de uma única simulação por ciclo, que têm menor custo evidentemente.

Os gráficos exibem um comportamento qualitativo semelhante e os resultados tendem a ser melhores quando se aumenta p_{in} . No entanto, mesmo utilizando-se os mesmos parâmetros, instâncias de redes diferentes conduziram a uma certa variação nos resultados. Nas Figuras 43 e 44, apresenta-se portanto um conjunto de resultados mais amplo, onde incrementa-se p_{in} progressivamente e aplica-se a técnica de remoção de arestas sobre um conjunto de 50 instâncias diferentes de rede para cada valor p_{in} .

Como esperado, o aumento de p_{in} torna mais fácil a tarefa de distinguir diferentes

Figura 42 – Valores ARI e de modularidade ao longo de diferentes ciclos, utilizando-se redes de diferentes probabilidades de conexão intra-comunidade p_{in} : 0,62, 0,64, 0,66 e 0,68. Cada rede tem 4 comunidades, cada comunidade formada por 200 vértices, e $\langle k \rangle \approx 10$. Cada gráfico é obtido rodando-se um conjunto de 10 simulações independentes por ciclo, e removendo-se 40 arestas ($\approx 1\%$) ao final de cada ciclo. O significado de cada gráfico é similar àqueles apresentados na Figura 40. Para todas as simulações, parâmetro $\alpha = 0, 1$.

comunidades. Nota-se também que há uma maior variabilidade nas regiões intermediárias dos gráficos, entre $p_{in} = 0,60$ e $0,70$. Devido ao comportamento razoavelmente similar exibido em todos os gráficos, torna-se um pouco difícil afirmar qual é a melhor abordagem. Esse é o motivo da Figura 44, na qual selecionam-se alguns valores de p_{in} e posicionam-se os resultados de diferentes abordagens lado-a-lado. Então torna-se claro que a aplicação do processo de remoção de arestas conduz a resultados consideravelmente melhores. Particularmente, ao invés utilizar as componentes conexas, o emprego dos algoritmos CFG ou Louvain após modificar a rede pela remoção das arestas parece ser a escolha mais segura. É ainda possível selecionar o ciclo ótimo por meio da análise da modularidade, como discutido anteriormente, o que pode levar a resultados ainda melhores.

Figura 43 – Resultados obtidos para um amplo conjunto de diferentes redes. Para cada probabilidade de conexão intra-comunidade p_{in} , emprega-se um conjunto de 50 redes distintas. Cada rede tem 4 comunidades, cada comunidade formada por 200 vértices, e $\langle k \rangle \approx 10$. Uma única simulação por ciclo é realizada. Além disso, 40 arestas ($\approx 1\%$) são removidas ao final de cada ciclo. Sumarizam-se os resultados pelos seguintes valores de precisão: menor, 1º quartil, mediana, 3º quartil e maior. A linha contínua e mais espessa representa a mediana. Os gráficos são dispostos pela aplicação dos algoritmos CFG e Louvain na rede original, antes que qualquer aresta tenha sido removida (resultados comparativos, mostrados em vermelho), e pela aplicação desses mesmos algoritmos após 30 e 40 ciclos de remoção (mostrados em azul). Os resultados provenientes das componentes conexas são também apresentados. Para todas as simulações, parâmetro $\alpha = 0,1$.

Figura 44 – Boxplots destacando os diferentes resultados para certas probabilidades p_{in} . Trata-se rigorosamente dos mesmos resultados mostrados na Figura 43, tomados em $p_{in} = 0,60, 0,65$ e $0,70$. Os resultados em azul são obtidos após aplicar o processo de remoção de arestas, diferentemente daqueles em vermelho, que fornecem resultados comparativos.

5.2.4 Complexidade Computacional

A abordagem baseada na remoção de arestas apresentada neste capítulo é, sob certo aspecto, semelhante àquela baseada na medida de *betweenness*, proposta por Newman e Girvan (2004). O conceito de entropia de aresta mede o quão diferente é um par de vértices em função do desalinhamento dos vetores-velocidade. Já o conceito de *betweenness* mede a relevância de uma aresta em função do número de caminhos mínimos que incluem tal aresta. Ambos os conceitos acabam tentando identificar arestas que conectam comunidades distintas. Uma deficiência considerável da abordagem baseada em *betweenness* é, no entanto, seu custo computacional cúbico. Precisamente, para uma rede de m arestas e n vértices, a complexidade de tempo é $O(m^2n)$, ou $O(n^3)$ para redes esparsas, nas quais $m \sim n$ (CLAUSET; NEWMAN; MOORE, 2004; NEWMAN; GIRVAN, 2004; LANCICHINETTI; FORTUNATO, 2009; FORTUNATO, 2010). Isso faz necessário que se tenha uma discussão do custo computacional demandado pela abordagem iterativa de remoção de arestas aqui apresentada.

Quatro variáveis são relevantes para nossa análise da complexidade de tempo:

1. o número n de vértices;
2. o número m de arestas;
3. a quantidade total de passos t_{total} por simulação;
4. e o número de ciclos de remoção de arestas n_{rounds} que é necessário para se obter uma partição adequada da rede.

A dimensionalidade dos vetores-velocidade é constante e bem pequena — somente três dimensões são empregadas nos experimentos —, portanto não é relevante para a análise. Nossa hipótese é que tanto t_{total} quanto n_{rounds} são invariantes, independentemente do quão grande a rede seja.

No caso de se remover somente uma aresta por ciclo, n_{rounds} toma a ordem de m . Entretanto, os experimentos apresentados na seção anterior indicam que é possível obter resultados satisfatórios mesmo removendo-se uma fração (uma porcentagem) de todas as arestas por ciclo, situação em que n_{rounds} se torna constante. Por exemplo, removendo-se aproximadamente 1% das arestas por ciclo requer menos de 50 ciclos para se atingir bons resultados. Além disso, com relação ao número de simulações por ciclo, não há nenhum ganho evidente ao se realizar mais de uma simulação. Mesmo que decida-se realizar um conjunto de simulações a fim de reduzir a influência da configuração aleatória inicial, tal número de simulações é ainda relativamente pequeno e não cresce com n nem m .

Com o intuito de fornecer uma evidência que suporte nossa hipótese — tanto t_{total} quanto n_{rounds} não crescem com o tamanho da rede —, apresentam-se aqui alguns resultados sobre redes 10 vezes maiores que aquelas estudadas na seção anterior. Nas Figuras 45 and 46, apresentam-se

Figura 45 – Evolução da entropia numa rede 10 vezes maior que aquelas estudadas na seção anterior: 4 comunidades, cada uma formada por 2000 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$. A população de valores de entropia foi obtida de 10 simulações independentes. Gráfico da esquerda: a evolução da entropia separada entre arestas intra e inter-comunidades. Gráfico da direita: no caso da mediana, por exemplo, mostra-se a razão $\frac{\text{mediana inter-comunidade}}{\text{mediana intra-comunidade}}$. Similarmente, mostram-se as razões para o 3º quartil e para o maior valor. Ambos os gráficos representam simulações no primeiro ciclo, ou seja, antes que qualquer aresta tenha sido removida. Parâmetro $\alpha = 0,1$.

Figura 46 – Resultados obtidos numa rede 10 vezes maior que aquelas estudadas na seção anterior: 4 comunidades, cada uma formada por 2000 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$. Cada gráfico é obtido pela realização de 10 simulações independentes por ciclo, e removendo-se 100 arestas ($\approx 0,25\%$) ou 400 arestas ($\approx 1\%$) ao final de cada ciclo. O significado de cada gráfico é similar àqueles já apresentados na seção anterior. Para todas as simulações, parâmetro $\alpha = 0,1$.

resultados detalhados sobre uma única instância de rede, ao passo que resultados mais significativos sobre uma população de 50 redes distintas são apresentados na Figura 47. Como esperado, o aumento do tamanho da rede não afeta nem o número de passos por simulação nem o número de ciclos de remoção necessários para se obter resultados satisfatórios. Consequentemente, pode-se estimar que, para uma rede arbitrariamente grande, tanto t_{total} quanto n_{rounds} permanecem prati-

Figura 47 – Resultados obtidos sobre uma população de 50 redes distintas. Cada rede é construída com os mesmos parâmetros daquela da Figura 46: 4 comunidades, cada uma formada por 2000 vértices, $\langle k \rangle \approx 10$ e $p_{\text{in}} = 0,60$. Apresentam-se os resultados de se remover 100 arestas ($\approx 0,25\%$) ou 400 arestas ($\approx 1\%$) por ciclo, obtidos com a aplicação do algoritmo CFG após diferentes números de ciclos de remoção, como também apresentam-se resultados comparativos da aplicação do CFG ou do Louvain na rede original, sem que nenhuma aresta seja removida. Uma única simulação por ciclo é realizada, e o parâmetro $\alpha = 0,1$ para toda simulação.

camente os mesmos, ao menos não crescem linearmente com o tamanho da rede. Nas Figuras 45 e 46, tanto a evolução da entropia quanto os valores ARI adquirem formas similares àquelas dos resultados sobre redes de menor porte, que são apresentadas nas seções anteriores.

Uma vez que todos os cálculos na Equação 5.1 (Seção 5.1.1) ocorrem sobre o conjunto de arestas, a complexidade de tempo para uma única iteração dinâmica t é $O(m)$. Sabendo que t_{total} não cresce com o tamanho da rede, a simulação inteira continua demandando $O(m)$. Além disso, ao final de cada ciclo (uma única simulação ou um pequeno conjunto de simulações), é necessário selecionar as arestas de maior entropia, algo que requer a ordenação do conjunto de arestas e normalmente demanda $O(m \log m)$. E sabendo que n_{rounds} também não cresce com o tamanho da rede, conclui-se que todo o processo de remoção de arestas tem complexidade de tempo quasilinear: $O(m \log m)$ se $m \gg n$, ou simplesmente $O(n \log n)$ para redes esparsas.

Entretanto, é importante enfatizar que o custo $O(m \log m)$ se dá apenas sob o ponto de vista assintótico, quando as redes se tornam bem grandes. Para redes de pequeno e médio porte, o custo causado por t_{total} e n_{rounds} se torna sensível. Além disso, no caso de se aplicar um algoritmo auxiliar para obter a partição final da rede, o custo de tal algoritmo também deve ser levado em conta. A maioria dos exemplos tratados neste trabalho empregam o algoritmo CFG que, para redes esparsas, normalmente roda em tempo quasilinear: $O(n \log^2 n)$ (CLAUSET; NEWMAN; MOORE, 2004).

Surpreendentemente, os resultados das Figuras 46 e 47 não apenas sustentam nossa hipótese — tanto t_{total} quanto n_{rounds} não crescem com o tamanho da rede —, mas também apresentam uma melhoria muito grande quando se emprega a abordagem de remoção de arestas aqui tratada. Quando aplicados sobre a rede original, ou seja, sem o processo de remoção de

arestas, os resultados dos algoritmos CFG e Louvain são tão ruins que seus boxplots praticamente desaparecem em torno de zero.

5.2.5 Critério de Parada do Sistema Dinâmico

Como discutido na Seção 5.2.1, ao invés de se aproximarem cada vez mais de zero, as entropias das arestas se dispõem em torno de um valor constante chamado *machine epsilon*. Pouco antes do sistema entrar nessa configuração estável em torno de *machine epsilon*, as entropias das arestas inter-comunidades já se tornaram, em geral, maiores que as intra-comunidades. No entanto, foi dito que a interrupção do processo dinâmico nesse ponto — antes de atingir *machine epsilon* — conduz a resultados muito inferiores. Nas Figuras 48 and 49, são apresentados alguns resultados onde as simulações são interrompidas em $t = 800$, imediatamente antes do estado estável (com relação ao primeiro ciclo, pois isso muda com o passar dos ciclos). Compare a Figura 48 com a Figura 37. Aqui, as simulações seguem as mesmas condições, exceto por serem mais curtas agora.

Observando-se os gráficos *inter-/intra-community entropy ratio*, embora os valores correspondentes à mediana e ao 3º quartil estão acima de 1 como esperado, o razão correspondente à maior entropia aparece abaixo de 1 no ciclo 10, ou seja, a entropia inter-comunidade máxima se torna menor que a intra-comunidade máxima. Além do ciclo 10, essa é uma observação comum em diversos outros ciclos (não mostrados aqui). Isso sugere que a seleção das arestas de maior entropia para remoção deixa de ser efetiva, exceto nos primeiros ciclos.

Além disso, em $t = 800$, o sistema já se mostra razoavelmente estável — os gráficos *inter-/intra-community entropy ratio* não demonstram nenhuma tendência de grande mudança. Isso sugere que as diferenças nos resultados finais não são uma simples questão de esperar mais (esperar até $t = 5000$, por exemplo), mas as diferenças têm sim algo a ver com a configuração em torno do valor *machine epsilon* que o conjunto de entropias toma. Portanto, entender por que as simulações mais longas conduzem a resultados muito melhores é algo que ainda merece futura investigação.

5.2.6 Experimentos com Comunidades Desbalanceadas

Todos os experimentos apresentados até agora foram realizados com redes balanceadas, cujas comunidades têm todas o mesmo tamanho. A fim de prover exemplos mais realísticos, são introduzidos agora alguns resultados com redes cujas comunidades têm diferentes quantidades de vértices. Tais resultados são mostrados nas Figuras 50 e 51.

Embora a probabilidade de conexão intra-comunidade p_{in} seja maior que a maioria dos exemplos apresentados até agora — 0,65 aqui contra 0,60 na maioria dos exemplos anteriores —, nota-se agora uma considerável queda na qualidade das partições. Ainda assim, comparado com algoritmos tradicionais como CFG ou Louvain aplicados na rede original, a técnica de

Figura 48 – Evolução da entropia sobre a rede mostrada na Figura 36: 4 comunidades, cada uma formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,60$. A população dos valores de entropia é obtida de 10 simulações independentes. Aqui, as simulações seguem as mesmas condições daquelas apresentadas na Figura 37, exceto por serem mais curtas agora ($t = 800$). Gráficos superiores: a evolução da entropia separada entre arestas intra e inter-comunidades. Gráficos inferiores: no caso da mediana, por exemplo, mostra-se a razão $\frac{\text{mediana inter-comunidade}}{\text{mediana intra-comunidade}}$. Similarmente, mostram-se as razões para o 3º quartil e para o maior valor. Gráficos à esquerda: simulações no primeiro ciclo, ou seja, antes que qualquer aresta tenha sido removida. Gráficos à direita: simulações no ciclo número 10, ou seja, após 9 ciclos de remoção de arestas.

Figura 49 – Resultados obtidos rodando-se cada simulação até $t = 800$. Do lado esquerdo, resultados detalhados numa única instância de rede: a mesma rede mostrada na Figura 36. Do lado direito, resultados numa população de 50 redes distintas. As caixas azuis são resultados da aplicação do algoritmo CFG após um certo número de ciclos de remoção de arestas. Cada rede tem 4 comunidades, cada comunidade formada por 200 vértices, $\langle k \rangle \approx 10$ e $p_{\text{in}} = 0,60$. 10 simulações independentes por ciclo são realizadas e 40 arestas ($\approx 1\%$) são removidas ao final de cada ciclo. Parâmetro $\alpha = 0,1$.

Figura 50 – Resultados com uma rede desbalanceada formada por quatro comunidades, cada uma contendo 800, 400, 200 e 100 vértices, $\langle k \rangle \approx 10$ e $p_{\text{in}} = 0,65$. À esquerda, a rede de entrada. À direita, o resultado da aplicação do algoritmo CFG após 74 ciclos de remoção de arestas. As comunidades resultantes são representadas por cores diferentes. 10 simulações independentes por ciclo são realizadas e 37 arestas ($\approx 0,5\%$) são removidas ao final de cada ciclo. Parâmetro $\alpha = 0,1$.

Figura 51 – Resultados obtidos com redes desbalanceadas formadas por quatro comunidades, cada uma contendo 800, 400, 200 e 100 vértices, $\langle k \rangle \approx 10$ e $p_{in} = 0,65$. À esquerda, resultados detalhados com uma única instância de rede: a mesma rede mostrada na Figura 50. À direita, resultados com uma população de 50 redes distintas. Os resultados em azul são obtidos pela aplicação do algoritmo CFG após certo número de ciclos de remoção. 10 simulações independentes por ciclo são realizadas nos resultados à esquerda, e uma única simulação por ciclo nos resultados à direita. 37 arestas ($\approx 0,5\%$) são removidas ao final de cada ciclo. Parâmetro $\alpha = 0,1$.

remoção iterativa de arestas oferece uma melhoria significativa, especialmente considerando os valores ARI numa população de 50 redes distintas, que são mostrados por meio dos boxplots da Figura 51. Similarmente ao que acontece com as redes balanceadas, os valores ARI mais altos parecem se estabelecer nas mesmas regiões onde a modularidade das componentes conexas atingem seu primeiro máximo local.

Vértices em pequenas comunidades têm à sua disposição um grande número de outros vértices em comunidades maiores para distribuir suas conexões, fazendo com que as conexões inter-comunidades se tornem bem mais esparsas que as intra-comunidades. Vértices em grandes comunidades, por outro lado, possuem poucos vértices de outras comunidades à disposição, consequentemente enfraquecendo as fronteiras entre diferentes comunidades. Olhando para a Figura 50, isso talvez explique por que a menor comunidade permanece intacta enquanto a maior se quebra em quatro grandes grupos: vértices vermelhos provenientes da menor comunidade, roxos da segunda menor, azuis da terceira menor, e verdes dela própria. Na Figura 52, exibem-se as componentes conexas obtidas após 50 e 54 ciclos. Uma grande transformação é vista num intervalo de poucos ciclos: toda a rede se divide em duas grandes componentes. Novamente, enquanto a menor comunidade permanece intacta, a maior se parte em dois grupos distintos.

Figura 52 – Componentes conexas após certo número de ciclos de remoção de arestas. Trata-se do mesmo processo cujos valores de precisão são apresentados no lado esquerdo da Figura 51. Diferentes cores denotam componentes distintas.

CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho de doutorado teve como objetivo geral mostrar que a auto-organização promovida por um grande grupo de partículas num sistema dinâmico apresenta bom desempenho na solução de uma ampla classe de problemas do aprendizado de máquina. Nesse sentido, a partir do desenvolvimento de diferentes técnicas e seu estudo em cenários diversos, entende-se que a expectativa foi atingida.

Uma característica interessante ao longo deste trabalho de doutorado foi partir de modelos complicados e, a partir do entendimento desses modelos, evoluir para modelos mais simples com igual ou até melhor poder de abrangência. Por exemplo, o modelo semissupervisionado apresentado no Capítulo 3, cheio de detalhes e relativamente complicado, evoluiu para modelos não-supervisionados capazes de resolver tarefas semelhantes sem fazer uso de itens rotulados. A técnica de segmentação de imagens, apresentada no Capítulo 4, é um exemplo desses modelos não-supervisionados. Levando-se também em conta os materiais apresentados à frente nos Apêndices A e B — materiais ainda preliminares, por isso deixados como apêndices —, pode-se comparar as duas técnicas de detecção de comunidades, uma apresentada no Apêndice B e outra no Capítulo 5: a primeira (Apêndice B) exige movimentação dos dados e é restrita a agrupamento ou redes espaciais, enquanto a segunda (Capítulo 5), desenvolvida depois e mais simples, apenas alinhamento, sem movimentação, parece ser capaz de resolver um escopo maior de tarefas. Contudo, vale notar que as duas técnicas são diferentes em custo computacional — a primeira exige somente um ciclo, uma simulação —, podendo haver, portanto, vantagens e desvantagens em cada uma.

Em todos os cenários estudados, mostrou-se que a configuração inicial aleatória das partículas pode conduzir a diferentes resultados em diferentes simulações. Essa característica, que em princípio pode parecer negativa, representa um potencial muito importante: revelar regiões de incerteza. No caso do aprendizado semissupervisionado, revelam-se áreas de intensa disputa pela propagação de rótulos de diferentes classes. Já no caso da segmentação de imagens,

diferentes interpretações pode haver para uma mesma cena.

Além da própria riqueza promovida pela natureza descentralizada e auto-organizável das técnicas aqui desenvolvidas, houve também um grande esforço em manter tais técnicas suficientemente eficientes para que pudessem ser aplicadas em grandes conjuntos de dados. Um método de particionamento do espaço de atributos foi desenvolvido e introduzido já no Capítulo 3, fazendo com que a técnica semissupervisionada ali apresentada tivesse um custo assintótico sub-quadrático.

Com o exemplo de aplicação apresentado no Capítulo 4, é possível ver que a técnica baseada na dinâmica coletiva de partículas exhibe propriedades interessantes quando aplicada à segmentação de imagens. Resultados consistentes são alcançados para uma extensa faixa de parâmetros. Entretanto, mais que trazer um novo algoritmo de segmentação, com este trabalho pretende-se trazer ideias capazes de motivar novos estudos e aplicações de modelos dinâmicos auto-organizáveis na solução de problemas complexos. Além disso, importantes propriedades podem ser reveladas analisando-se estados intermediários ao longo do processo dinâmico, por exemplo, o quão resistente é um grupo de partículas a ser quebrado em subgrupos, ou então onde se localiza a fronteira entre grupos distintos, uma vez que a ruptura pode ocorrer em diferentes locais. Na Seção 4.2.2, é discutida a variabilidade de resultados para uma mesma configuração de parâmetros, que é causada pela atribuição aleatória dos vetores-velocidade iniciais. Tal variabilidade indica diferentes interpretações de uma mesma imagem, e pode portanto ser explorada como um meio de se expressar incerteza.

Com relação ao Capítulo 5, por meio de um amplo conjunto de resultados experimentais apresentados na Seção 5.2, é possível constatar que o sistema dinâmico de alinhamento de partículas junto com a técnica iterativa de remoção de arestas são capazes de melhorar consideravelmente os resultados de métodos tradicionais como CFG ou Louvain. A natureza descentralizada e auto-organizável do modelo dinâmico é robusta, portanto aplicável numa ampla variedade de redes.

O conceito de entropia de aresta lá definida é, sob certo aspecto, semelhante ao conceito de *betweenness*. Arestas de alta entropia provavelmente conectam comunidades distintas. Entretanto, o custo $O(n^3)$ (em redes esparsas) do método proposto por Newman e Girvan (2004) torna proibitivo seu uso em redes de larga escala. Por outro lado, argumenta-se que o processo de remoção de aresta baseado na entropia é assintoticamente quasilinear, $O(n \log n)$, tornando promissor seu emprego na detecção de comunidades em redes de larga escala.

Uma questão importante é levantada nas Seções 5.2.1 e 5.2.5, com relação à dependência que o sucesso da técnica tem na configuração estável em torno do valor *machine epsilon*. Entender o que exatamente acontece em tal configuração é, porém, matéria que ainda merece futura investigação. Modificações no modelo dinâmico de alinhamento apresentado na Seção 5.1.1 poderiam resolver essa questão. Numa direção similar, quando observam-se os gráficos “inter/intra-entropy-ratio” da Figura 48, é possível ver que selecionar rigorosamente as

arestas de maior entropia pode não ser a melhor abordagem, portanto pode-se também explorar melhores meios de selecionar as arestas a serem removidas.

REFERÊNCIAS

ABRAHAM, A.; DAS, S.; ROY, S. Soft computing for knowledge discovery and data mining. In: _____. [S.l.]: Springer Verlag, 2007. cap. Swarm intelligence algorithms for data clustering. Citado nas páginas 30 e 31.

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SüSSTRUNK, S. Slic superpixels compared to state-of-the-art superpixel methods. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 11, p. 2274–2281, 2012. Citado nas páginas 63 e 67.

ADAMS, R.; BISCHOF, L. Seeded region growing. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 16, n. 6, p. 641–647, 1994. Citado na página 61.

ALBERT, R.; BARABÁSI, A. L. Statistical mechanics of complex networks. **Rev. Modern Phys.**, v. 74, n. 1, p. 47–97, 2002. Citado na página 32.

ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: The MIT Press, 2004. (Adaptive computation and machine learning series). Citado nas páginas 21 e 34.

ALPERT, S.; GALUN, M.; BRANDT, A.; BASRI, R. Image segmentation by probabilistic bottom-up aggregation and cue integration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 2, p. 315–327, 2012. Citado na página 61.

ARENAS, A.; DÍAZ-GUILERA, A.; PÉREZ-VICENTE, C. J. Synchronization reveals topological scales in complex networks. **Physical Review Letters**, v. 96, n. 11, 2006. Citado na página 86.

BARRAT, A.; BARTHÉLEMY, M.; VESPIGNANI, A. **Dynamical processes on complex networks**. [S.l.]: Cambridge University Press, 2008. ISBN: 978-0-521-87950-7. Citado na página 32.

BELKIN, M.; NIYOGI, P. Laplacian eigenmaps for dimensionality reduction and data representation. **Neural Computation**, v. 15, p. 1373–1396, 2003. Citado na página 58.

_____. Convergence of laplacian eigenmaps. **preprint, short version NIPS 2008**, 2008. Citado na página 58.

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, p. 2399–2434, 2006. Citado nas páginas 36 e 58.

BENGIO, Y.; DELALLEAU, O.; ROUX, N. L. Semi-supervised learning. In: _____. [S.l.]: The MIT Press, 2006. cap. Label propagation and quadratic criterion. ISBN: 978-0-262-03358-9. Citado na página 35.

BEUCHER, S.; MEYER, F. Mathematical morphology in image processing. In: _____. [S.l.]: CRC Press, 1993. cap. The morphological approach to segmentation: the watershed transformation, p. 433–481. Citado na página 81.

- BEZDEK, J.; HATHAWAY, R. Vot: A tool for visual assessment of (cluster) tendency. In: **Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN '02)**. [S.l.]: IEEE, 2002. p. 2225–2230. Citado na página 130.
- BISHOP, C. M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006. (Information science and statistics). Citado na página 21.
- BLATT, M.; WILSEMAN, S.; DOMANY, E. Data clustering using a model granular magnet. **Neural Computation**, v. 9, n. 8, p. 1805–1842, 1997. Citado nas páginas 30 e 31.
- BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. **Journal of Statistical Mechanics: Theory and Experiment**, v. 2008, n. 10, 2008. Citado nas páginas 85, 86, 91, 96, 97 e 140.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. **Swarm intelligence: from natural to artificial systems**. [S.l.]: Oxford University Press, 1999. ISBN: 0-19-513159-2. Citado nas páginas 22 e 23.
- BREVE, F.; ZHAO, L.; QUILES, M. Uncovering overlap community structure in complex networks using particle competition. In: **Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence**. [S.l.]: Springer, 2009. p. 619–628. Citado na página 86.
- CAPOCCI, A.; SERVEDIO, V. D. P.; CALDARELLI, G.; COLAIORI, F. Detecting communities in large networks. **Physica A: Statistical Mechanics and its Applications**, v. 352, n. 2–4, p. 669–676, 2005. Citado na página 85.
- CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. (Ed.). **Semi-supervised learning**. [S.l.]: The MIT Press, 2006. ISBN: 978-0-262-03358-9. Citado nas páginas 15, 22, 32, 33, 35, 57, 58, 59 e 60.
- CHAPELLE, O.; SINDHWANI, V.; KEERTHI, S. S. Optimization techniques for semi-supervised support vector machines. **Journal of Machine Learning Research**, v. 9, p. 203–233, 2008. Citado na página 58.
- CHAPELLE, O.; ZIEN, A. Semi-supervised classification by low density separation. In: **Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics**. [S.l.: s.n.], 2005. p. 57–64. Citado na página 58.
- CHATÉ, H.; GINELLI, F.; GRÉGOIRE, G.; PERUANI, F.; RAYNAUD, F. Modeling collective motion: variations on the vicsek model. **The European Physical Journal B**, v. 64, p. 451–456, 2008. Citado nas páginas 22 e 28.
- CLAUSET, A.; NEWMAN, M. E. J.; MOORE, C. Finding community structure in very large networks. **Physical Review E**, v. 70, n. 6, 2004. Citado nas páginas 85, 86, 91, 96, 97, 103, 105 e 140.
- CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, p. 273–297, 1995. Citado na página 58.
- COSTA, L. da F.; RODRIGUES, F. A.; TRAVIESO, G.; VILLAS BOAS, P. R. Characterization of complex networks: A survey of measurements. **Advances in Physics**, v. 56, n. 1, p. 167–242, 2007. Citado nas páginas 85, 86, 97 e 140.

- COUPRIE, C.; GRADY, L.; NAJMAN, L.; TALBOT, H. Power watershed: a unifying graph-based optimization framework. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 33, n. 7, p. 1384–1399, 2011. Citado na página 61.
- COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE Transactions on Information Theory**, v. 13, p. 21–27, 1967. Citado na página 58.
- CRUCITTI, P.; LATORA, V.; PORTA, S. Centrality measures in spatial networks of urban streets. **Physical Review E**, v. 73, n. 3, 2006. Citado nas páginas 133 e 135.
- CUI, X.; GAO, J.; POTOK, T. E. A flocking based algorithm for document clustering analysis. **Journal of Systems Architecture**, v. 52, p. 505–515, 2006. Citado na página 31.
- CUPERTINO, T. H.; GUELERI, R.; ZHAO, L. A semi-supervised classification technique based on interacting forces. **Neurocomputing**, v. 127, p. 43–51, 2014. Citado nas páginas 30, 31, 40, 58, 59 e 60.
- DONETTI, L.; MUÑOZ, M. A. Detecting network communities: a new systematic and efficient algorithm. **Journal of Statistical Mechanics: Theory and Experiment**, v. 2004, n. 10, 2004. Citado na página 85.
- DORIGO, M.; STÜTZLE, T. **Ant colony optimization**. [S.l.]: The MIT Press, 2004. ISBN: 0-262-04219-3. Citado na página 23.
- El-Zahhar, M. M. A semi-supervised learning approach for soft labeled data. In: **Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications (ISDA)**. [S.l.: s.n.], 2010. p. 1136–1141. ISBN: 978-1-4244-8134-7. Citado na página 24.
- ENGELBRECHT, A. P. **Fundamentals of computational swarm intelligence**. [S.l.]: Wiley, 2005. ISBN: 978-0-470-09191-3. Citado nas páginas 22, 23 e 31.
- EXPERT, P.; EVANS, T. S.; BLONDEL, V. D.; LAMBIOTTE, R. Uncovering space-independent communities in spatial networks. **Proceedings of the National Academy of Sciences of the United States of America (PNAS)**, v. 108, n. 19, p. 7663–7668, 2011. Citado nas páginas 133 e 135.
- FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011. ISBN: 978-8521618805. Citado nas páginas 21, 22 e 34.
- FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Efficient graph-based image segmentation. **International Journal of Computer Vision**, v. 59, n. 2, p. 167–181, 2004. Citado nas páginas 61, 81 e 82.
- FORTUNATO, S. Community detection in graphs. **Physics Reports**, v. 486, n. 3–5, p. 75–174, 2010. Citado nas páginas 85, 86, 97 e 103.
- FREEMAN, L. C. A set of measures of centrality based on betweenness. **Sociometry**, v. 40, n. 1, p. 35–41, 1977. Citado na página 86.
- GAO, Z.-K.; YANG, Y.-X.; FANG, P.-C.; JIN, N.-D.; XIA, C.-Y.; HU, L.-D. Multi-frequency complex network from time series for uncovering oil-water flow structure. **Scientific Reports**, v. 5, n. 8222, 2015. Citado na página 85.

- GINELLI, F.; PERUANI, F.; BÄR, M.; CHATÉ, H. Large-scale collective properties of self-propelled rods. **Physical Review Letters**, v. 104, n. 18, 2010. Citado nas páginas 22 e 28.
- GOLDBERG, D. What every computer scientist should know about floating-point arithmetic. **ACM Computing Surveys**, v. 23, n. 1, p. 5–48, 1991. Citado na página 92.
- GRADY, L. Random walks for image segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 28, n. 11, p. 1768–1783, 2006. Citado nas páginas 61 e 81.
- GUELERI, R. A. **Agrupamento de dados baseado em comportamento coletivo e auto-organização**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação — Universidade de São Paulo, São Carlos, SP, Brazil, 2013. Citado nas páginas 42, 43, 44 e 45.
- GUELERI, R. A.; CUPERTINO, T. H.; CARVALHO, A. C. de; ZHAO, L. A flocking-like technique to perform semi-supervised learning. In: **Proceedings of the “2014 International Joint Conference on Neural Networks – World Congress on Computational Intelligence (IJCNN/WCCI 2014), Beijing, China”**. [S.l.: s.n.], 2014. p. 1579–1586. Citado nas páginas 48, 53, 55 e 57.
- GUELERI, R. A.; ZHAO, L. Agrupamento de dados baseado em dinâmica de troca de energia. In: **Anais do XXXI Congresso da Sociedade Brasileira de Computação (CSBC 2011)**. [s.n.], 2011. p. 418–429. ISSN: 2175-2761. Disponível em: <<http://www.dimap.ufrn.br/csbc2011>>. Citado nas páginas 30 e 31.
- _____. An energy exchanging mechanism for data clustering. In: **Proceedings of the 2012 Brazilian Symposium on Neural Networks (SBRN 2012)**. [S.l.: s.n.], 2012. p. 31–36. ISBN: 978-0-7695-4823-4. Citado nas páginas 30 e 31.
- _____. Image segmentation via superpixels self-organized motion. In: **Proceedings of the “4th Symposium on Knowledge Discovery, Mining and Learning (KDMiLe 2016), Recife, PE, Brazil”**. [S.l.: s.n.], 2016. p. 44–51. Citado na página 70.
- GUELERI, R. A.; ZHENG, Q.; ZHANG, J.; ZHAO, L. Swarm intelligence: from concepts to applications. In: _____. [S.l.]: IET, 2018. cap. Image segmentation via flocking-like particle dynamics. In press. Citado nas páginas 65, 66, 68, 69, 71, 72, 73, 74, 76, 77, 78, 79, 80, 82, 83 e 84.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Springer, 2001. (Springer series in statistics). Citado nas páginas 21 e 34.
- HEEMSKERK, E. M.; TAKES, F. W. The corporate elite community structure of global capitalism. **New Political Economy**, v. 21, n. 1, p. 90–118, 2016. Citado na página 85.
- HIGHAM, N. J. **Accuracy and stability of numerical algorithms**. [S.l.]: SIAM, 2002. Citado na página 92.
- HUBERT, L.; ARABIE, P. Comparing partitions. **Journal of Classification**, v. 2, n. 1, p. 193–218, 1985. Citado nas páginas 69 e 96.
- HÜLLERMEIER, E. Fuzzy methods in machine learning and data mining: status and prospects. **Fuzzy Sets and Systems**, n. 156, p. 387–406, 2005. Citado na página 24.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Computing Surveys**, v. 31, n. 3, p. 264–323, 1999. Citado nas páginas 21 e 22.

JOACHIMS, T. Transductive learning via spectral graph partitioning. In: **Proceedings of the International Conference on Machine Learning**. [S.l.: s.n.], 2003. p. 290–297. Citado na página 58.

_____. Semi-supervised learning. In: _____. [S.l.]: The MIT Press, 2006. cap. Transductive support vector machines. Citado nas páginas 34 e 58.

JONSSON, P. F.; CAVANNA, T.; ZICHA, D.; BATES, P. A. Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. **BMC Bioinformatics**, v. 7, n. 2, 2006. Citado na página 85.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **IEEE International Conference on Neural Networks**. [S.l.: s.n.], 1995. v. 4, p. 1942–1948. Citado na página 31.

KENNEDY, J.; EBERHART, R. C. **Swarm intelligence**. [S.l.]: Academic Press, 2001. Citado nas páginas 22 e 31.

LANCICHINETTI, A.; FORTUNATO, S. Community detection algorithms: a comparative analysis. **Physical Review E**, v. 80, n. 5, 2009. Citado nas páginas 86 e 103.

_____. Limits of modularity maximization in community detection. **Physical Review E**, v. 84, n. 6, 2011. Citado nas páginas 86 e 97.

LANCICHINETTI, A.; KIVELÄ, M.; SARAMÄKI, J.; FORTUNATO, S. Characterizing the community structure of complex networks. **PLoS ONE**, v. 5, n. 8, 2010. Citado na página 85.

LLOYD, S. P. Least squares quantization in pcm. **IEEE Transactions on Information Theory**, v. 28, n. 2, p. 129–137, 1982. Citado nas páginas 128 e 131.

MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: **Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability**. [S.l.: s.n.], 1967. p. 281–297. Citado nas páginas 128 e 131.

MENON, G. I. **Active matter**. 2010. ArXiv:1003.2032v1. Citado nas páginas 22 e 28.

MINH, H. Q.; SINDHWANI, V. Vector-valued manifold regularization. In: **Proceedings of the 28th International Conference on Machine Learning**. [S.l.: s.n.], 2011. Citado na página 58.

MISHRA, S.; TUNSTRØM, K.; COUZIN, I. D.; HUEPE, C. Collective dynamics of self-propelled particles with variable speed. **Physical Review E**, v. 86, n. 1, 2012. Citado nas páginas 22, 23, 86 e 88.

MITCHELL, T. M. **Machine learning**. [S.l.]: McGraw-Hill, 1997. (McGraw-Hill series in computer science). Citado na página 21.

MITROVIĆ, M.; TADIĆ, B. Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities. **Physical Review E**, v. 80, n. 2, 2009. Citado na página 85.

NAGAI, K. H.; SUMINO, Y.; MONTAGNE, R.; ARANSON, I. S.; CHATÉ, H. Collective motion of self-propelled particles with memory. **Physical Review Letters**, v. 114, n. 16, 2015. Citado nas páginas 22, 23, 86 e 88.

- NEWMAN, M. The structure and function of complex networks. **SIAM Rev.**, v. 45, n. 2, p. 167–256, 2003. Citado na página 32.
- NEWMAN, M. E. J.; GIRVAN, M. Finding and evaluating community structure in networks. **Physical Review E**, v. 69, n. 2, 2004. Citado nas páginas 85, 86, 97, 103 e 112.
- NGO, S.; GINELLI, F.; CHATÉ, H. Competing ferromagnetic and nematic alignment in self-propelled polar particles. **Physical Review E**, v. 86, n. 5, 2012. Citado nas páginas 22 e 28.
- NIGAM, K.; MCCALLUM, A.; MITCHELL, T. Semi-supervised learning. In: _____. [S.l.]: The MIT Press, 2006. cap. Semi-supervised text classification using EM. ISBN: 978-0-262-03358-9. Citado na página 34.
- NOCK, R.; NIELSEN, F. Statistical region merging. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 26, n. 11, p. 1452–1458, 2004. Citado na página 61.
- OHLANDER, R.; PRICE, K.; REDDY, D. R. Picture segmentation using a recursive region splitting method. **Computer Graphics and Image Processing**, v. 8, n. 3, p. 313–333, 1978. Citado na página 61.
- OLIVEIRA, T. B. S. **Clusterização de dados utilizando técnicas de redes complexas e computação bioinspirada**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, São Carlos, SP, Brazil, 2008. Citado nas páginas 30 e 31.
- OMRAN, M.; SALMAN, A.; ENGELBRECHT, A. P. Image classification using particle swarm optimization. In: **Conference on simulated evolution and learning**. [S.l.: s.n.], 2002. v. 1, p. 370–374. Citado nas páginas 30 e 31.
- PENG, B.; ZHANG, L.; ZHANG, D. A survey of graph theoretical approaches to image segmentation. **Pattern Recognition**, v. 46, n. 3, p. 1020–1038, 2013. Citado na página 61.
- PESHKOV, A.; ARANSON, I. S.; BERTIN, E.; CHATÉ, H.; GINELLI, F. **Simple and faithful nonlinear field equations for aligning self-propelled rods**. 2012. ArXiv:1207.5751v1. Citado nas páginas 22 e 28.
- QUILES, M. G.; MACAU, E. E. N.; RUBIDO, N. Dynamical detection of network communities. **Scientific Reports**, v. 6, n. 25570, 2016. Citado nas páginas 85 e 86.
- QUILES, M. G.; ZHAO, L.; ALONSO, R. L.; ROMERO, R. A. F. Particle competition for complex network community detection. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, v. 18, n. 3, 2008. Citado na página 86.
- RAMASWAMY, S. The mechanics and statistics of active matter. **Annual Review of Condensed Matter Physics**, v. 1, p. 323–345, 2010. Citado nas páginas 22 e 28.
- RAND, W. M. Objective criteria for the evaluation of clustering methods. **Journal of the American Statistical Association**, v. 66, n. 336, p. 846–850, 1971. Citado nas páginas 69 e 128.
- REICHARDT, J.; BORNHOLDT, S. Detecting fuzzy community structures in complex networks with a potts model. **Physical Review Letters**, v. 93, n. 21, 2004. Citado na página 86.

REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. **ACM SIGGRAPH Computer Graphics**, v. 21, n. 4, p. 25–34, 1987. Citado nas páginas 29, 31, 47, 126 e 134.

_____. **Steering behaviors for autonomous characters**. 1999. Web page. Disponível em: <<http://www.red3d.com/cwr/steer/gdc99/>>. Citado nas páginas 29, 30, 31, 47, 126 e 134.

ROSENBERG, D. S.; SINDHWANI, V.; BARTLETT, P. L.; NIYOGI, P. A kernel for semi-supervised learning with multi-view point cloud regularization. **IEEE Signal Processing Magazine**, 2009. Citado na página 58.

SEEGER, M. **Learning with labeled and unlabeled data**. [S.l.], 2002. Citado nas páginas 22 e 32.

SHIH, F. Y.; CHENG, S. Automatic seeded region growing for color image segmentation. **Image and Vision Computing**, v. 23, n. 10, p. 877–886, 2005. Citado na página 61.

SILVA, T. C.; ZHAO, L. Detecting overlapping structures via network-based competitive learning. In: **Proceedings of the 2012 IEEE World Congress on Computational Intelligence (WCCI 2012)**. [S.l.: s.n.], 2012. Citado nas páginas 30, 31 e 32.

_____. Network-based stochastic semisupervised learning. **IEEE Transactions on Neural Networks and Learning Systems**, v. 23, n. 3, p. 451–466, 2012. Citado nas páginas 22, 30, 31, 32, 35, 39, 40 e 41.

_____. Stochastic competitive learning in complex networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 23, n. 3, p. 385–398, 2012. Citado nas páginas 30, 31, 32, 85 e 86.

_____. **Machine learning in complex networks**. [S.l.]: Springer International Publishing, 2016. ISBN: 978-3-319-17289-7 (print) 978-3-319-17290-3 (online). Citado nas páginas 85 e 86.

SILVA, T. C.; ZHAO, L.; CUPERTINO, T. H. Handwritten data clustering using agents competition in networks. **Journal of Mathematical Imaging and Vision**, 2012. Citado nas páginas 30, 31 e 32.

SINDHWANI, V.; BELKIN, M.; NIYOGI, P. Semi-supervised learning. In: _____. [S.l.]: The MIT Press, 2006. cap. The geometric basis of semi-supervised learning. ISBN: 978-0-262-03358-9. Citado na página 35.

SINDHWANI, V.; NIYOGI, P.; BELKIN, M. Beyond the point cloud: from transductive to semi-supervised learning. In: **Proceedings of the 22nd International Conference on Machine Learning**. [S.l.: s.n.], 2005. p. 824–831. Citado na página 58.

STORATH, M.; WEINMANN, A. Fast partitioning of vector-valued images. **SIAM Journal on Imaging Sciences**, v. 7, n. 3, p. 1826–1852, 2014. Citado nas páginas 81 e 82.

STORATH, M.; WEINMANN, A.; FRIKEL, J.; UNSER, M. Joint image reconstruction and segmentation using the potts model. **Inverse Problems**, v. 31, n. 2, 2015. Citado nas páginas 81 e 82.

SUMINO, Y.; NAGAI, K. H.; SHITAKA, Y.; TANAKA, D.; YOSHIKAWA, K.; CHATE, H.; OIWA, K. Large-scale vortex lattice emerging from collectively moving microtubules. **Nature**, v. 483, p. 448–452, 2012. Citado nas páginas 22, 28, 86 e 88.

- SUN, J.; BOYD, S.; XIAO, L.; DIACONIS, P. The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem. **SIAM Review**, v. 48, p. 681–699, 2006. Citado na página 58.
- THIEMANN, C.; THEIS, F.; GRADY, D.; BRUNE, R.; BROCKMANN, D. The structure of borders in a small world. **PLoS ONE**, v. 5, n. 11, 2010. Citado na página 85.
- TOBIAS, O. J.; SEARA, R. Image segmentation by histogram thresholding using fuzzy sets. **IEEE Transactions on Image Processing**, v. 11, n. 12, p. 1457–1465, 2002. Citado na página 61.
- TONER, J. Birth, death, and flight: a theory of malthusian flocks. **Physical Review Letters**, v. 108, n. 8, 2012. Citado nas páginas 22, 28, 86 e 88.
- _____. Reanalysis of the hydrodynamic theory of fluid, polar-ordered flocks. **Physical Review E**, v. 86, n. 3, 2012. Citado nas páginas 22 e 28.
- TONER, J.; TU, Y. Long-range order in a two-dimensional dynamical xy model: how birds fly together. **Physical Review Letters**, v. 75, n. 23, p. 4326–4329, 1995. Citado nas páginas 22 e 28.
- _____. Flocks, herds, and schools: a quantitative theory of flocking. **Physical Review E**, v. 58, n. 4, p. 4828–4858, 1998. Citado nas páginas 22 e 28.
- van der Walt, S.; SCHÖNBERGER, J. L.; NUNEZ-IGLESIAS, J.; BOULOGNE, F.; WARNER, J. D.; YAGER, N.; GOUILLART, E.; YU, T. scikit-image: image processing in python. **PeerJ**, v. 2, n. e453, 2014. Citado na página 81.
- VERRI, F. A. N.; URIO, P. R.; ZHAO, L. Network unfolding map by vertex-edge dynamics modeling. **IEEE Transactions on Neural Networks and Learning Systems**, PP, n. 99, 2016. Citado na página 86.
- VICSEK, T.; CZIRÓK, A.; Ben-Jacob, E.; COHEN, I.; SHOCHET, O. Novel type of phase transition in a system of self-driven particles. **Physical Review Letters**, v. 75, n. 6, p. 1226–1229, 1995. Citado nas páginas 22, 23, 28, 29, 30, 48, 86, 88 e 134.
- VICSEK, T.; ZAFEIRIS, A. Collective motion. **Physics Reports**, v. 517, p. 71–140, 2012. Citado nas páginas 22, 23, 86, 88 e 134.
- WANG, F. A general learning framework using local and global regularization. **Pattern Recognition**, v. 43, p. 3120–3129, 2010. Citado nas páginas 22, 32, 35, 37 e 38.
- WANG, X.; TANG, Y.; MASNOU, S.; CHEN, L. A global/local affinity graph for image segmentation. **IEEE Transactions on Image Processing**, v. 24, n. 4, p. 1399–1411, 2015. Citado na página 61.
- WATTS, D. J.; STROGATZ, S. H. Collective dynamics of small-world networks. **Nature**, v. 393, p. 440–442, 1998. Citado na página 140.
- XU, R.; Wunsch II, D. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, v. 16, n. 3, p. 645–678, 2005. Citado nas páginas 21 e 22.
- XU, R.; Wunsch II, D. C. **Clustering**. [S.l.]: Wiley - IEEE Press, 2009. (IEEE Press Series on Computational Intelligence). Citado na página 34.

YANG, B.; LIU, J. Discovering global network communities based on local centralities. **ACM Transactions on the Web**, v. 2, n. 1, 2008. Citado na página 85.

YANG, M. S. A survey of fuzzy clustering. **Mathl. Comput. Modelling**, v. 18, n. 11, p. 1–16, 1993. Citado na página 24.

ZHANG, A. **Protein intraction networks**. [S.l.]: Cambridge University Press, 2009. ISBN: 978-0-521-88895-0. Citado na página 85.

ZHAO, L.; Damiance Jr., A. P. G.; CARVALHO, A. C. P. L. F. Lecture notes in computer science: advances in natural computation. In: _____. [S.l.]: Springer, 2005. v. 3610, n. 422, cap. A self-organized network for data clustering. Citado nas páginas 30 e 31.

ZHONG, C.; ARISONA, S. M.; HUANG, X.; BATTY, M.; SCHMITT, G. Detecting the dynamics of urban structure through spatial network analysis. **International Journal of Geographical Information Science**, v. 28, n. 11, p. 2178–2199, 2014. Citado na página 85.

ZHOU, D.; SCHÖLKOPF, B. Semi-supervised learning. In: _____. [S.l.]: The MIT Press, 2006. cap. Discrete regularization. ISBN: 978-0-262-03358-9. Citado na página 35.

ZHU, X. **Semi-supervised learning literature survey**. [S.l.], 2008. Relatório técnico (Computer Sciences TR 1530, University of Wisconsin-Madison). Citado nas páginas 22 e 32.

ZHU, X.; GOLDBERG, A. B. **Introduction to semi-supervised learning**. [S.l.]: Morgan & Claypool, 2009. (Synthesis lectures on artificial intelligence and machine learning). ISBN: 9781598295474. Citado nas páginas 22 e 32.

ADAPTAÇÃO AO CENÁRIO NÃO-SUPERVISIONADO

A técnica dinâmica desenvolvida para aprendizado semissupervisionado e apresentada no Capítulo 3 (Aprendizado Semissupervisionado via Dinâmica Coletiva de Partículas) contém muitos detalhes e, conforme previsto no plano de desenvolvimento deste trabalho de doutorado, versões mais fundamentais — e mais simples — deveriam ser modeladas e analisadas. O entendimento das versões mais simples daria então suporte ao entendimento das versões mais sofisticadas. Especificamente, a principal simplificação consistiria em eliminar os itens rotulados, que desempenham papéis especiais. Têm-se, conseqüentemente, o caso não-supervisionado, em que todos os itens desempenham papéis similares. Tal versão mais fundamental foi então modelada e estudada. Seus resultados, embora ainda preliminares, são apresentados neste capítulo.

A.1 Descrição do Modelo

Consideremos o seguinte modelo, que rege a movimentação de cada partícula $\mathbf{x}_i$, a cada iteração (tempo) t :

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \alpha_1(t) \cdot \hat{\mathbf{v}}_i(t) \quad (\text{A.1})$$

$$\mathbf{v}_i(t+1) = \hat{\mathbf{v}}_i(t) + \alpha_2 \cdot \hat{\mathbf{v}}_i^{\text{chg}}(t) \quad (\text{A.2})$$

$$\begin{aligned} \mathbf{v}_i^{\text{chg}} = \sum_{\mathbf{x}_j} \left(\text{align} \cdot (\hat{\mathbf{v}}_j - \hat{\mathbf{v}}_i) \cdot \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2} + \right. \\ \left. + \text{approx} \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \cdot \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2} + \right. \\ \left. + \text{sep} \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \cdot \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^3} \right) \end{aligned} \quad (\text{A.3})$$

A velocidade inicial $\mathbf{v}_i(0)$ de todas as partículas é atribuída aleatoriamente. $\hat{\mathbf{v}}_i$ e $\hat{\mathbf{v}}_i^{\text{chg}}$ denotam, respectivamente, os vetores unitários $\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$ e $\frac{\mathbf{v}_i^{\text{chg}}}{\|\mathbf{v}_i^{\text{chg}}\|}$. Especialmente, na Equação A.1, $\hat{\mathbf{v}}_i(t)$ define o movimento de velocidade constante, característica comum na modelagem de sistemas auto-propulsionados e que se repete ao longo deste trabalho de doutorado. $\|\mathbf{x}_i - \mathbf{x}_j\|$ é a distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$, de modo que quanto menor a distância separando duas partículas, maior será a influência de uma sobre a outra. α_1 e α_2 são parâmetros, e definem a intensidade com que a posição e a velocidade das partículas variam, respectivamente. As constantes align, approx e sep são também parâmetros e denotam, respectivamente, os termos de alinhamento, aproximação e separação das partículas, característica baseada no modelo de *boids* proposto por Reynolds (1987), Reynolds (1999).

A.2 Resultados Experimentais

Três versões do modelo apresentado na seção anterior foram estudadas:

1. α_1 tem valor constante, portanto independente de t . Trata-se, na verdade, da concepção original do projeto, onde tanto α_1 quanto α_2 são constantes tomadas como parâmetros do modelo. Essa versão apresentou dificuldades em seu estudo, pois valores elevados de α_1 conduziam a resultados muito ruins, enquanto valores baixos demandavam um tempo muito grande para que o sistema atingisse uma configuração em que os dados de classes distintas ficassem bem separados. A explicação é razoavelmente simples: no início, as partículas têm velocidades aleatórias; elas precisam de um tempo para se alinharem e, consequentemente, para que grupos consistentes possam ser formados. Valores pequenos de α_1 resolvem isso mas, por outro lado, implicam que as partículas se movimentarão muito lentamente, chegando a tornar inviável o emprego desse modelo. Abandonou-se, portanto, o estudo dessa versão. A solução foi justamente fazer α_1 bem pequeno no início e, conforme o tempo passa e as partículas se alinham, seu valor vai sendo incrementado. Trata-se das duas versões apresentadas a seguir, que adotam $\alpha_1 = 10^{\text{exp}} \cdot t^3$. Atentar para o expoente exp, um dos principais parâmetros analisados nos experimentos. O valor fixo de α_2 foi determinado empiricamente para diferentes conjuntos de dados.
2. Somente o termo de alinhamento (align) considerado. Assim, faz-se align = 1, approx = 0 e sep = 0.

Figura 53 – Ilustração do processo dinâmico modelado, em diferentes estágios. Cada item inicia com direção aleatória. Conjunto de dados de 1000 itens. Apenas termo de alinhamento. Dados bidimensionais (2 atributos), mas embutidos em espaço de 5 dimensões, para reduzir chance de colisão. Apesar das cores usadas para visualização, a técnica é não-supervisionada, portanto não “conhece” nenhuma classe.

3. Todos os três termos considerados. Faz-se, novamente, $\text{align} = 1$. Mas agora, estuda-se o impacto de diferentes valores do parâmetro approx . Já o parâmetro sep é determinado em tempo de execução, e leva em conta o valor de approx e a densidade dos dados.

Os resultados a seguir referem-se somente à segunda e à terceira versões, visto que a primeira foi abandonada por se apresentar inviável.

Na Figura 53, é ilustrado o comportamento do processo dinâmico modelado. Os dados possuem dois atributos — duas dimensões — mas são embutidos num espaço de cinco dimensões, para aumentar o grau de liberdade e evitar colisões indesejadas, conforme discutido à frente. Apesar das cores usadas para visualização, provenientes das classes dos dados gerados artificialmente, importante reforçar que a técnica dinâmica aqui proposta não “conhece” nenhuma dessas classes. Todos os itens são tratados indistintamente.

A seguinte metodologia foi adotada nos experimentos:

1. Dois conjuntos artificiais de dados foram selecionados: (1) duas gaussianas de mesmo tamanho dispostas lado-a-lado; e (2) dois arcos entrelaçados (este último ilustrado na Figura 53).

2. As duas últimas versões do modelo foram selecionadas: (1) somente o termo de alinhamento; e (2) todos os três termos.
3. No caso do modelo que considera todos os três termos — alinhamento, aproximação e separação —, dois valores distintos do parâmetro *approx* foram selecionados. Esses valores, considerados representativos, foram determinados empiricamente.
4. Os conjuntos de dados estudados são bidimensionais — dois atributos. Assim, foram estudados dois casos: (1) o espaço por onde as partículas se movem é também bidimensional; e (2) três dimensões adicionais foram empregadas, totalizando cinco dimensões, o que oferece maior grau de liberdade aos dados e reduzem drasticamente a chance de haver colisões indesejadas. De fato, já era esperado que o emprego de somente duas dimensões conduziria a resultados ruins, algo que foi confirmado nos experimentos, e tais resultados são aqui omitidos. Somente os resultados para cinco dimensões, resultados estes mais relevantes, são aqui apresentados.
5. Variou-se o expoente (*exp*) por uma ampla gama de valores. Valores pequenos — mais negativos — oferecem mais tempo para as partículas se alinharem, antes de iniciar movimento mais intenso.
6. Para cada uma dessas configurações, 30 simulações foram executadas, totalizando cerca de 18.000 simulações. Para cada simulação, diferentes estados do sistema foram registrados em arquivo, para posterior análise.
7. O sistema foi avaliado em diferentes estágios de evolução. Para isso, estabeleceu-se o conceito de *raio*. O raio é definido em função da maior das distâncias entre a posição de uma partícula e a origem do sistema de coordenadas do espaço. Por isso, no início do processo, é importante que o conjunto de dados seja centralizado em torno da origem. O raio vale 1 para o estado inicial, e um raio valendo 2, por exemplo, significa que os dados estão 2 vezes mais expandidos do que estavam no início. Trata-se simplesmente de um critério de parada, visto que o sistema evolui indefinidamente. Raios valendo 2, 4, 8 e 16 foram registrados.
8. O modelo aqui apresentado não agrupa diretamente os dados, mas sim os transforma de modo a realçar seus padrões. Assim, algoritmos de agrupamento propriamente ditos devem ser aplicados no final. Dos algoritmos testados, o *k-médias* (MACQUEEN, 1967; LLOYD, 1982) é ao mesmo tempo o mais simples e o que conduziu a resultados mais consistentes. Os resultados apresentados a seguir foram obtidos com o *k-médias*.
9. Por fim, o índice Rand (RAND, 1971) foi tomado como critério que mede a qualidade da partição obtida. Outros índices foram também tomados, mas não evidenciaram diferenças significativas nos resultados, em relação ao índice Rand.

Figura 54 – Índice Rand em função do expoente. Diferentes raios (tempo de evolução) representados por diferentes cores: 1 ($t = 0$, dados em sua disposição original) – cinza; 2 – vermelha; 4 – verde; 8 – azul; e 16 – magenta. Para cada configuração, 30 simulações. Exibem-se o 1º quartil, a mediana e o 3º quartil. Espaço de 5 dimensões. K-médias aplicado no final do processo.

Figura 55 – Mapa de distâncias entre as partículas para diferentes valores de expoente e diferentes raios (tempo de evolução). Raio igual a 1 representa os dados em sua disposição original ($t = 0$). Ponto mais escuro significa maior distância (normalizada). Conjunto de dados semelhante ao da Figura 53 (1000 itens). Somente termo de alinhamento. Espaço de 5 dimensões. Apenas 200 itens são mostrados, tomados aleatoriamente. 1–100: itens da classe 1; 101–200: classe 2.

Alguns resultados foram selecionados e sumarizados na Figura 54. Já na Figura 55, mostra-se o mapa de distâncias entre as partículas em diferentes estágios de evolução do processo dinâmico. Trata-se de um mapa similar ao VAT (*Visual Assessment of (Cluster) Tendency*), proposto por Bezdek e Hathaway (2002), mas aqui fazendo uso do conhecimento das classes para ordenar os itens. Em duas das figuras (exp. -18, raios 4 e 16), aparece um item completamente fora do padrão. Trata-se de algum item, provavelmente de fronteira, proveniente de uma certa classe mas que tomou a direção dos itens da classe oposta.

A.3 Discussão dos Resultados

A partir da análise dos resultados apresentados na seção anterior, algumas conclusões importantes puderam ser tomadas:

- Lembrando que aplicou-se o k-médias para o agrupamento definitivo dos dados, não há ganho para o conjunto formado por duas gaussianas idênticas (*Data 1*), já que o k-médias sozinho já atua muito bem em dados desse tipo (MACQUEEN, 1967; LLOYD, 1982). No entanto, o ganho é substancial para conjuntos mais complicados como aquele ilustrado na Figura 53 (*Data 2*).
- Embora o emprego de apenas duas dimensões (quantidade exata de atributos) conduza a resultados insatisfatórios (não mostrados aqui), a adição de somente mais três dimensões já traz um ganho muito grande na qualidade dos resultados. Como discutido, isso é devido à colisões indesejadas que se evitam com mais dimensões.
- Conforme evidenciado na Figura 54, o emprego dos termos de aproximação e separação degrada os resultados, especialmente no segundo conjunto de dados (*Data 2*). O modelo que emprega somente o termo de alinhamento, além de ser mais simples, apresenta resultados bons e estáveis. A principal razão é que o modelo completo, que inclui os termos de aproximação e separação, simplesmente tem dificuldades em agrupar dados dispostos em formato alongado. Ele só funciona bem com dados mais concentrados em torno de um ponto — uma gaussiana. Além disso, foi observado que, especialmente quando as partículas adquirem altas velocidades, os grupos se tornam instáveis e se fragmentam.
- Embora os resultados da Figura 54 (*Data 2, only align.*) mostrem-se muito bons para todos os valores de expoente, os resultados do mapa de distâncias da Figura 55 mostram que há uma diferença significativa nos padrões formados. Especialmente para o expoente valendo -14 (valor muito alto), a primeira classe (itens 1–100) se torna fragmentada, dividindo-se em dois grupos. Um comportamento mais consistente é alcançado com valores intermediários de expoente. De fato, é de se esperar que valores altos de expoente deem menos tempo para os dados se alinharem, e portanto identifica somente grupos de tamanho pequeno. Já expoentes baixos permitem a formação de grupos maiores. No entanto, valores extremamente baixos permitirão que todos os dados se alinhem, formando um único grupo, algo evidentemente indesejável. Conclui-se que o valor do expoente, que é um parâmetro, está diretamente associado à quantidade e ao tamanho dos grupos que se deseja obter.

REDES COM TOPOLOGIA ESPACIAL E AGRUPAMENTO DE DADOS

Similarmente à técnica desenvolvida no Capítulo 5 (Detecção de Comunidades via Remoção Iterativa de Arestas), o modelo aqui apresentado desvincula os dados de seu espaço de atributos. Como consequência, o modelo pode encontrar sua aplicação em redes, que são objetos puramente topológicos, cujos vértices não têm uma posição definida em nenhum espaço de atributos. São apenas vértices conectados entre si por meio de arestas. No entanto, a rede tratada aqui ainda possui uma motivação espacial, portanto este trabalho se restringe a redes de topologia espacial (CRUCITTI; LATORA; PORTA, 2006; EXPERT *et al.*, 2011).

Como é discutido mais à frente neste capítulo, um conjunto de dados vetoriais pode, de maneira simples, ser transformado numa rede k-NN. Nesse sentido, pode-se transformar um problema de agrupamento de dados num caso mais geral de detecção de comunidades em redes, restrito neste caso a redes de topologia espacial. Portanto, pode-se dizer que a técnica aqui proposta além de ser uma técnica de detecção de comunidades em redes espaciais, é também uma técnica de agrupamento de dados.

Os resultados apresentados neste capítulo são ainda preliminares e merecem mais estudo. Resultados mais completos sobre o assunto de detecção de comunidades em redes são apresentados no Capítulo 5, porém em redes mais gerais em que não há nenhuma motivação espacial.

B.1 Descrição do Modelo

Dada uma rede não-direcionada, livre de autoconexões e arestas múltiplas, faz-se cada vértice corresponder a uma partícula $\mathbf{x}_i$ em algum espaço multidimensional. As arestas definem a vizinhança de cada $\mathbf{x}_i$, ou seja, aquelas partículas com as quais $\mathbf{x}_i$ pode interagir diretamente. Essa rede de interação não muda ao longo do processo dinâmico, embora dois itens inicialmente

vizinhos deixarão de interagir um com o outro quando a distância entre eles for suficientemente grande. Uma vez que a rede não muda, o método de particionamento do espaço utilizado em capítulos anteriores deixa de ser necessário.

Inicialmente, cada partícula $\mathbf{x}_i(0)$ é posicionada aleatoriamente no espaço. Cada coordenada segue uma distribuição normal de probabilidade, com média 0 e desvio-padrão $\sigma = 1$. Três dimensões foram empregadas nos resultados aqui apresentados. Testes empíricos mostraram que duas dimensões conduzem a resultados indesejáveis, enquanto três ou mais dimensões funcionam de modo consistente. A cada partícula $\mathbf{x}_i$, associa-se também uma velocidade inicial aleatória $\mathbf{v}_i(0)$, descrita por um vetor unitário que aponta para uma direção aleatória.

O processo dinâmico de movimentação de partículas é então regido pelo seguinte sistema:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \alpha_1(t) \cdot \hat{\mathbf{v}}_i(t) \quad (\text{B.1})$$

$$\mathbf{v}_i(t+1) = \hat{\mathbf{v}}_i(t) + \alpha_2 \cdot \hat{\mathbf{v}}_i^{\text{chg}}(t) \quad (\text{B.2})$$

$$\mathbf{v}_i^{\text{chg}} = \frac{1}{k_i} \cdot \sum_{\mathbf{x}_j | d_{ij} < d_0} (\hat{\mathbf{v}}_j - \hat{\mathbf{v}}_i) \quad (\text{B.3})$$

$$\alpha_1 = 10^{\text{exp}} \cdot t^3 \quad (\text{B.4})$$

Embora omitido para simplificação, com ocorre na Equação B.3, todas as variáveis $\mathbf{x}$, $\mathbf{v}$ e $\mathbf{v}^{\text{chg}}$ são sempre em função de t . Por conseguinte, $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ também é função de t .

Tanto $\hat{\mathbf{v}}_i$ quanto $\hat{\mathbf{v}}_i^{\text{chg}}$ denotam vetores unitários, ou seja, $\hat{\mathbf{v}}_i = \mathbf{v}_i / \|\mathbf{v}_i\|$ e $\hat{\mathbf{v}}_i^{\text{chg}} = \mathbf{v}_i^{\text{chg}} / \|\mathbf{v}_i^{\text{chg}}\|$. Na Equação B.1, isso garante que todas as partículas se movam com a mesma velocidade a cada instante t . Essa é uma característica comumente modelada nos sistemas de partículas autopropulsionadas, e que tem sido referência desde o início deste trabalho de doutorado (REYNOLDS, 1987; VICSEK *et al.*, 1995; REYNOLDS, 1999; VICSEK; ZAFEIRIS, 2012).

α_1 e α_2 definem o quão rápido as posições e as velocidades são atualizadas, respectivamente. Notar que α_1 é, na verdade, uma função de t , e é atualizado de acordo com a Equação B.4, onde define-se também um parâmetro chamado *exp*. Como consequência, o valor de *exp* vem a definir o valor de α_1 . Trata-se de um parâmetro importante e bastante estudado em nossos experimentos. Uma vez que as direções de velocidade são atribuídas aleatoriamente no início, é desejável que as partículas desempenhem um movimento bem lento, dando tempo para que o conjunto de vetores-velocidade se torne mais organizado. Conforme o tempo avança, os vetores-velocidade se tornam mais alinhados e mudam mais lentamente, então é possível permitir que as partículas se movimentem a passos mais largos. Isso é exatamente o que a Equação B.4 faz, aumentando progressivamente o valor de α_1 .

Na Equação B.3, a somatória se dá sobre todos os vizinhos $\mathbf{x}_j$, desde que a distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$ ($d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$) não supere um limiar de distância d_0 . O limiar $d_0 = 8$ tem sido

empregado nos experimentos, cujo valor é arbitrário mas em concordância com a distribuição normal de probabilidade com $\sigma = 1$ definida no início do processo.

Observar que, diferentemente de alguns dos modelos apresentados em capítulos anteriores, aqui a intensidade da interação entre dois itens não diminui (progressivamente) com o aumento da distância entre eles. Isso acontece porque não só o conjunto de velocidades iniciais $\mathbf{v}_i(0)$ é definido aleatoriamente, mas aqui é também aleatória a disposição dos próprios itens $\mathbf{x}_i(0)$. Fazer com que distâncias pequenas promovam forte interação num sistema cujas partículas estão aleatoriamente distribuídas é indesejável, pois isso descaracterizaria os padrões de conectividade da rede que se quer realçar. Por outro lado, não havendo interferência da distância no cálculo da interação faria com que todos os itens acabassem tomando praticamente a mesma direção de movimento. A existência do limiar d_0 portanto interrompe a interação num estágio intermediário, após padrões úteis serem formados e antes que todo o conjunto se torne completamente alinhado. A partir desse limiar, grupos distintos tomam direções também distintas, realçando cada vez mais a diferença entre eles. Pelas Figuras 56, 57 e 58, ilustram-se tanto o processo dinâmico aqui modelado, quanto a diferença entre aplicar ou não aplicar o limiar d_0 . Na Figura 56, a disposição inicial aleatória das partículas. Já nas Figuras 57 e 58, respectivamente a ausência e a presença do limiar d_0 . Nota-se a “ruptura” entre diferentes grupos quando se emprega o limiar de distância de interação d_0 .

k_i é o grau do vértice i , consequentemente a quantidade de vizinhos de $\mathbf{x}_i$. Desse modo, $1/k_i$ é um termo de normalização, evitando que itens de grau muito elevado se alinhem mais rápido que os outros de menor vizinhança.

Como critério de parada, é necessário primeiramente definir o conceito de *raio*. A cada instante t , o *raio* é definido como sendo a distância entre a origem do espaço e a partícula mais distante dessa origem. Assim, o processo dinâmico é interrompido quando atinge-se um certo raio. Seu valor é arbitrário, mas deve ser grande o suficiente para permitir que o conjunto de partículas exiba algum padrão espacial útil, decorrente do processo de auto-organização. Um raio de 256 vezes o valor do raio inicial foi então empregado como critério de parada.

B.2 Experimentos: Quatro Comunidades com Diferentes Densidades

Experimentos foram realizados com o modelo descrito na seção anterior. Na Figura 59, apresenta-se um conjunto de dados vetoriais formados por dois atributos (duas dimensões). Grupos de diferentes densidades e uma certa sobreposição entre esses grupos tornam a tarefa um tanto desafiadora. Na mesma Figura 59, apresenta-se também uma rede k-NN (*k-Nearest Neighbors*, ou k vizinhos mais próximos), neste caso 10-NN, formada a partir dos dados. A própria natureza de construção da rede k-NN conduz a uma rede com topologia espacial (CRUCITTI; LATORA; PORTA, 2006; EXPERT *et al.*, 2011).

Figura 56 – Disposição inicial aleatória das partículas. Em cima: os vetores-velocidade são mostrados. Embaixo: as arestas da rede são mostradas. É a mesma rede de 3000 vértices apresentada à frente, na Figura 59. Cada cor representa uma comunidade, mas como trata-se de um cenário não-supervisionado, o processo dinâmico evidentemente não conhece *a priori* nenhuma comunidade. O espaço por onde as partículas se movem tem três dimensões, mas somente a projeção nas duas primeiras é mostrada.

Figura 57 – Disposição final das partículas, sem o emprego do limiar d_0 . Trata-se da disposição das partículas quando se atinge um raio 256 vezes o raio inicial. A rede de entrada e a configuração inicial das partículas são rigorosamente as mesmas mostradas na Figura 56. Novamente, as cores representam comunidades da rede, mas tal informação é ignorada pelo processo dinâmico. Parâmetros $\exp = -11$ e $\alpha_2 = 0,5$.

Figura 58 – Disposição final das partículas, empregando-se o limiar $d_0 = 8$. Trata-se da disposição das partículas quando se atinge um raio 256 vezes o raio inicial. A rede de entrada e a configuração inicial das partículas são rigorosamente as mesmas mostradas na Figura 56. Novamente, as cores representam comunidades da rede, mas tal informação é ignorada pelo processo dinâmico. Parâmetros $\exp = -11$ e $\alpha_2 = 0,5$.

Figura 59 – Dados utilizados nos experimentos aqui apresentados. Em cima: conjunto de dados vetoriais em duas dimensões (dois atributos). Há quatro grupos cujas regiões têm o mesmo tamanho, mas contendo diferentes quantidades de itens, consequentemente tendo diferentes densidades: 200, 400, 800 e 1600 itens, totalizando 3000 itens. Notar que há uma pequena sobreposição entre pares de grupos adjacentes, aumentando a densidade na região de intersecção e adicionando um pouco mais de dificuldade à tarefa de separação dos grupos. Embaixo: rede k-NN formada a partir dos dados, neste caso, 10-NN.

A técnica aqui trabalhada considera a rede como um objeto puramente topológico, ou seja, os vértices não têm nenhuma localização em nenhum espaço de atributos. No entanto, como mostrado na Figura 59, um conjunto de dados vetoriais pode, de maneira simples, ser transformado numa rede k-NN. Nesse sentido, pode-se transformar um problema de agrupamento de dados num caso mais geral de detecção de comunidades em redes, restrito neste caso a redes de topologia espacial. Portanto, pode-se dizer que a técnica aqui proposta além de ser uma técnica de detecção de comunidades em redes espaciais, é também uma técnica de agrupamento de dados.

Nas Figuras 60 e 61, apresentam-se os resultados da aplicação de dois algoritmos famosos da literatura: Cluster Fast Greedy (CFG) (CLAUSET; NEWMAN; MOORE, 2004) e Louvain (BLONDEL *et al.*, 2008). É notável a dificuldade que tais algoritmos apresentam em separar corretamente as comunidades presentes na rede.

Já nas Figuras 62 e 63, apresentam-se os resultados da técnica de movimentação de partículas proposta neste trabalho. Na Figura 62, mostra-se uma representação em forma de cores das distâncias entre as partículas no momento em que o processo dinâmico é interrompido. Na Figura 63, mostra-se a partição da rede obtida pela remoção das arestas que representam grandes distâncias. Resultados para diferentes valores de expoente (parâmetro exp) são exibidos. As comunidades encontradas refletem, com bastante precisão, o padrão original dos dados.

B.3 O Efeito “Small World”

O modelo dinâmico apresentado na Seção B.1 apresenta bons resultados quando aplicado a redes que exibem topologia espacial, como é o caso das redes k-NN. No entanto, nossos experimentos mostraram problemas ao se tentar aplicar o mesmo modelo a redes que exibem o chamado efeito “small world”, que são redes nas quais a maioria dos vértices pode ser alcançada a partir dos outros percorrendo-se um pequeno número de arestas (WATTS; STROGATZ, 1998; COSTA *et al.*, 2007). Nossa primeira hipótese residiu no termo $(\hat{\mathbf{v}}_j - \hat{\mathbf{v}}_i)$, na Equação B.3. Por um lado, o termo apresenta uma propriedade desejável de ser suave, ou seja, não apresentar transições abruptas quando um par de partículas vai se alinhando, assim evitando problemas relacionados à estabilidade. Por outro, ele faz com que partículas com direções de movimentação muito diferentes tenham interação muito forte, quando conectadas diretamente por uma aresta, o que poderia ser fonte de problemas justamente nas redes que apresentam o efeito “small world”.

Nesse sentido, uma modificação no modelo da Seção B.1 foi realizada:

$$\mathbf{v}_i^{\text{chg}} = \frac{1}{k_i} \cdot \sum_{\mathbf{x}_j | d_{ij} < d_0} \frac{\hat{\mathbf{v}}_j - \hat{\mathbf{v}}_i}{\|\hat{\mathbf{v}}_j - \hat{\mathbf{v}}_i\|} \quad (\text{B.5})$$

Desse modo, todo par de vetores-velocidade possui um mesmo grau de interação, não importando o quão alinhados eles estão. Adicionalmente, pode-se checar se $\hat{\mathbf{v}}_i$ e $\hat{\mathbf{v}}_j$ não são idênticos, assim ignorando tal cálculo de interação e evitando divisão por zero.

Figura 60 – Resultados obtidos com a aplicação do algoritmo Cluster Fast Greedy. Em cima: partição em quatro comunidades. Embaixo: partição de melhor modularidade. Cada cor representa uma comunidade encontrada.

Figura 61 – Resultados obtidos com a aplicação do algoritmo Louvain. Uma vez que esse algoritmo não retorna uma hierarquia de partições, não é possível solicitar um número específico de comunidades. Assim, apenas o resultado de melhor modularidade é exibido. Cada cor representa uma comunidade encontrada.

Experimentos que utilizam esse novo modelo foram realizados e apresentaram bons resultados. Na Figura 64, apresentam-se a rede empregada nos experimentos e os resultados da aplicação da técnica aqui proposta. Já na Figura 65, os resultados de outros algoritmos tradicionais da literatura são apresentados, a saber: os algoritmos CFG e Louvain.

Figura 62 – Representação em forma de cores das distâncias entre as partículas no momento em que o processo dinâmico é interrompido. Cada aresta toma uma cor que varia do ciano ao vermelho. Ciano representa um par de partículas que se mantiveram próximas uma da outra. Vermelho representa o contrário: um par de partículas que se afastaram bastante uma da outra, portanto trata-se de uma aresta que conecta duas comunidades distintas. Resultados empregando dois valores de expoente distintos são aqui mostrados (parâmetro $\exp$): $-10,5$ (em cima) e $-12,5$ (embaixo). Empregou-se também $\alpha_2 = 0,5$. Devido a uma certa sensibilidade à configuração aleatória inicial, o resultado é uma combinação de 50 execuções independentes.

Figura 63 – Resultados obtidos com a técnica proposta neste trabalho. A partir das distâncias tomadas pelas partículas em sua disposição final (Figura 62), a rede é particionada removendo-se as arestas que representam grandes distâncias, tomando-se um certo valor de limiar. Similarmente à Figura 62, resultados empregando dois valores de expoente distintos são aqui mostrados: $-10,5$ (em cima) e $-12,5$ (embaixo). Cada cor representa uma componente conexa da rede, ou seja, uma comunidade encontrada.

Figura 64 – Resultados da aplicação da técnica proposta neste trabalho sobre uma rede com efeito “small world”. Em cima: a rede original composta por duas comunidades, contendo 1000 vértices cada. Essa rede é formada por uma topologia espacial, similar àquela apresentada na seção anterior (Seção B.2), mas agora possui também um conjunto de arestas que introduz o efeito “small world”. No meio: representação em forma de cores das distâncias entre as partículas no momento em que o processo dinâmico é interrompido, similar à Figura 62. Valor de expoente utilizado: -14 . Embaixo: resultado obtido com a técnica proposta neste trabalho, removendo-se as arestas que representam grandes distâncias.

Figura 65 – Resultados obtidos com a aplicação de outros algoritmos da literatura. Em cima: Cluster Fast Greedy (CFG) com duas comunidades. No meio: CFG com partição de melhor modularidade. Embaixo: Louvain com partição de melhor modularidade. Cada cor representa uma comunidade encontrada.