

Universidade Estadual de Maringá

Departamento de Física

Alvaro Franco Martins

Ciência de Redes e Aprendizagem de
Máquina Aplicadas ao Estudo de Redes
Criminosas

Maringá, 22 de março de 2024.

Universidade Estadual de Maringá

Departamento de Física

Alvaro Franco Martins

Ciência de Redes e Aprendizagem de
Máquina Aplicadas ao Estudo de Redes
Criminosas

Tese de doutorado apresentada ao Programa
de Pós-Graduação em Física da Universidade
Estadual de Maringá.

Orientador: Prof. Dr. Haroldo Valentin Ribeiro

Maringá, 22 de março de 2024.

Resumo

Neste trabalho, adotamos abordagens de ciência de redes e aprendizagem de máquina para estudar propriedades e a dinâmica de redes criminosas, além de prever ligações e outras variáveis dessas redes. No Capítulo 1, apresentamos uma investigação abrangente sobre redes de corrupção relacionadas a escândalos na Espanha e no Brasil. Mostramos que as redes de corrupção desses dois países compartilham características estruturais e dinâmicas similares, como distribuições de grau, coeficientes de agrupamento e assortatividade próximos, além da presença de estruturas modulares. Também observamos um processo de crescimento marcado pela coalescência das componentes da rede devido à reincidência de alguns criminosos. Propomos um modelo simples que não apenas reproduz essas propriedades empíricas, mas também revela que as redes de corrupção operam perto de uma taxa crítica de reincidência abaixo da qual a rede é totalmente fragmentada e acima da qual é excessivamente conectada. No Capítulo 2, aplicamos técnicas de aprendizagem de máquina às redes de corrupção, bem como às redes de inteligência policial e de lavagem de dinheiro. Combinamos métodos de aprendizagem de representação de grafos e aprendizagem de máquina para revelar que as propriedades estruturais das redes estudadas podem ser utilizadas para prever parcerias criminosas, distinguir entre diferentes tipos de associações e também prever a quantidade de dinheiro trocado entre os agentes. Além disso, nossa abordagem se mostrou capaz de antecipar futuras conexões à medida que as redes de corrupção evoluem no tempo. No Capítulo 3, empregamos métodos de aprendizado profundo nas mesmas redes investigadas no Capítulo 2 visando obter melhores previsões sobre as mesmas variáveis. Exploramos o potencial das redes convolucionais de grafos para aprender padrões relacionados às relações criminosas e constatamos que modelos de aprendizado profundo baseados no algoritmo *GraphSAGE* são capazes de prever parcerias criminosas, distinguir entre tipos de associações e prever a quantidade de dinheiro trocado entre os agentes criminosos. Além disso, mostramos que é possível antecipar parcerias e a reincidência de criminosos durante o crescimento das redes de corrupção. Em geral, nossos modelos de aprendizado profundo superaram significativamente a abordagem apresentada no Capítulo 2.

Palavras-chave: Corrupção. Espanha. Brasil. Redes de Corrupção. Redes Criminosas. Lavagem de Dinheiro. Ciência de Redes. Sistemas Complexos. Ciência de Dados. Aprendizagem de máquina. Redes Neurais. Redes Convolucionais.

Abstract

In this work, we adopt network science and machine learning approaches to study properties and dynamics of criminal networks, as well as predict links and variables of these networks. In Chapter 1, we present a comprehensive investigation of corruption networks related to scandals in Spain and Brazil. We show that these corruption networks share similar structural and dynamical characteristics, such as degree distributions, clustering coefficients, and assortativity values, as well as the presence of modular structures. We also observe a growth process marked by the coalescence of network components due to a few recidivist agents. We propose a simple model that not only reproduces these empirical properties but also reveals that corruption networks operate around a critical recidivism rate below which the network becomes completely fragmented and above which it is overly connected. In Chapter 2, we apply machine learning techniques to corruption, criminal police intelligence, and criminal financial networks. We combine graph representation learning and machine learning methods to reveal that structural properties of these networks can be used to predict criminal partnerships, distinguish between different types of associations, and also predict the amount of money exchanged between agents. Furthermore, we use a similar approach to anticipate future connections as corruption networks grow over time. In Chapter 3, we employ deep learning methods on the same networks investigated in Chapter 2 to obtain better predictions for the same predictive tasks. We explore the potential of graph convolutional networks to learn patterns about criminal relationships and find that deep learning models based on the *GraphSAGE* algorithm can predict criminal partnerships, distinguish between types of associations, and predict the amount of money exchanged between agents. Additionally, we show that it is possible to anticipate partnerships and criminal recidivism during the growth of corruption networks. Overall, our deep learning models significantly outperform the approach presented in Chapter 2.

Keywords: Corruption. Spain. Brazil. Corruption Networks. Criminal Networks. Money Laundering. Network Science. Complex Systems. Data Science. Machine Learning. Neural Network. Convolutional Neural Network.

Sumário

Introdução	8
1 Universalidade de redes de corrupção política	11
1.1 Bases de dados	12
1.2 Análise quantitativa dos escândalos de corrupção	12
1.3 Redes de corrupção política	13
1.4 Evolução temporal da distribuição de grau	16
1.5 Dinâmica de crescimento das redes de corrupção	18
1.6 Comportamentos lineares nas redes de corrupção	20
1.7 Modelo para redes de corrupção	22
1.8 Robustez dos resultados	30
1.9 Conclusões	34
2 Aprendizagem de máquina aplicado a redes criminosas	36
2.1 Bases de dados	37
2.2 Prevendo parcerias em redes criminosas	37
2.3 Classificando a associação entre envolvidos da rede de inteligência policial .	41
2.4 Prevendo valores de transações bancárias na rede de crime financeiros . . .	44
2.5 Prevendo parcerias futuras nas redes de corrupção	46
2.6 Conclusões	50
3 Aprendizado profundo aplicado a redes criminosas	52
3.1 Prevendo parcerias em redes criminosas	53
3.2 Classificando a associação entre envolvidos da rede de inteligência policial .	57
3.3 Prevendo valores de transações bancárias na rede de crime financeiros . .	60
3.4 Prevendo parcerias futuras nas redes de corrupção	63
3.5 Prevendo a ocorrência de envolvidos reincidentes	66
3.6 Conclusões	70
Conclusões e perspectivas	71
A Conceitos de ciência de redes	73
A.1 Definição de redes	73
A.2 Grau e distribuição de grau	74
A.2.1 Grau	74
A.2.2 Distribuição de grau	74

A.3	Medidas estruturais	74
A.3.1	Densidade	74
A.3.2	Coeficiente de agrupamento	74
A.3.3	Assortatividade	75
A.3.4	Comprimento médio do caminho	75
A.4	O algoritmo <i>Infomap</i>	75
B	Conceitos de estatística	78
B.1	Distribuição acumulada	78
B.2	Método de máxima verossimilhança aplicado à distribuição exponencial . .	79
B.3	Método <i>bootstrap</i>	80
C	Métodos de aprendizagem estatística	81
C.1	Função <i>sigmoide</i>	82
C.2	Função <i>softmax</i>	82
C.3	Regressão logística	82
C.4	<i>k</i> -primeiros vizinhos	83
C.5	Acurácia	84
C.6	Matriz de Confusão	85
C.7	Coeficiente de determinação	86
C.8	Erro quadrático médio	87
C.9	Entropia cruzada	87
C.10	<i>Node2Vec</i>	88
C.11	<i>LINE</i>	91
C.12	Mercator	93
C.13	<i>UMAP</i>	94
C.14	GraphSAGE	96
	Referências bibliográficas	107

Introdução

Compreender a natureza coletiva e intrincada de corrupção política e de outros crimes organizados exige mais do que simples estatística. Em uma analogia com sistemas complexos [1–4], em que o todo é muitas vezes mais do que apenas a soma de suas partes, pode-se dizer que o sucesso de organizações criminosas não depende apenas das habilidades individuais dos criminosos, mas muito mais de sua capacidade de cooperar e criar estruturas capazes de proteger e esconder suas atividades ilegais. O uso da ciência da complexidade tem sido defendido por diferentes autores como uma abordagem ideal para investigar crime econômico, crime organizado e corrupção [5–8]. Nesse contexto, a ciência de redes [9, 10] tem um papel de destaque por possibilitar a descrição adequada das diferentes interações entre criminosos via uma ampla gama de ferramentas e métodos desenvolvidos nas últimas décadas [11].

Diversos trabalhos recentes demonstram a utilidade da ciência de redes para investigar redes criminosas, com exemplos que incluem detecção de cartel [12], risco de corrupção em contratos públicos [13], lavagem de dinheiro [14], identificação de políticos corruptos via redes de votação [15], redes de pedófilos da *dark web* [16], redes de conspiração entre empresas [17], estrutura modular de organizações criminosas [18], redes de corrupção política [19], redes de crime organizado [20], controlabilidade de redes criminosas [21], resiliência do narcotráfico [22], bem como redes de inteligência policial [23]. No entanto, apesar da fascinante pesquisa já realizada, existem lacunas importantes relacionadas à identificação de propriedades comuns e aspectos dinâmicos de redes criminosas, o que pode permitir desenvolver modelos simples que descrevam características fundamentais e forneçam informações úteis sobre o crime organizado.

Outro potencial do uso de redes complexas para tratar atividades criminosas é fornecer uma abordagem para realizar tarefas de previsões do comportamento criminoso futuro. Exemplos de tais tarefas incluem encontrar ligações não conhecidas (ou futuras) entre indivíduos e outras propriedades relacionadas às suas associações. Métodos de aprendizagem de máquina se tornaram prevalentes em investigações científicas em uma ampla gama de disciplinas, incluindo ciência de materiais [24, 25], química [26], física [27], biologia [28] e sociologia [29]. A proliferação recente dessas técnicas está intimamente relacionada ao rápido crescimento da quantidade de informações detalhadas sobre diversos sistemas, bem como ao desenvolvimento de abordagens de inteligência artificial capazes de lidar com os

mais variados tipos de dados. De fato, muitas disciplinas científicas estão cada vez mais dependentes de métodos capazes de extrair conhecimento útil de conjuntos de dados de grande escala e muitas vezes heterogêneos.

No entanto, ainda existem poucas tentativas de aplicar métodos de aprendizagem de máquina para prever propriedades estáticas e dinâmicas de redes criminosas [19, 30–32]. A escassez de tais estudos reflete os desafios de transformar informações de vértices e ligações em dados que possam ser usados por algoritmos de aprendizagem de máquina. Ao contrário de séries temporais ou imagens, que são organizadas em *arrays* ou estruturas bidimensionais, as redes possuem estruturas de dados mais complexas, com vértices e ligações indicando relacionamentos entre os vértices sem nenhuma associação espacial. Essa diferença é crucial porque as abordagens de aprendizagem de máquina que dependem de aspectos espaciais ou temporais, como redes neurais convolucionais [33] ou recorrentes [34], não são adequadas para redes.

Abordagens iniciais para extrair recursos de grafos focavam em combinações de estatísticas de rede, como medidas de centralidade e agrupamento, mas logo se mostraram limitadas devido à falta de generalização. Entretanto, métodos mais recentes são muito mais flexíveis e podem ser agrupados em duas categorias [35]: métodos tradicionais de incorporação de grafos e redes neurais de grafos. A primeira categoria inclui algoritmos de incorporação, como DeepWalk [36] e *Node2Vec* [37], que usam caminhadas aleatórias sobre as redes para gerar vetores associados aos vértices de tal maneira que vértices similares também possuam vetores similares no espaço de incorporação. A obtenção desses vetores representa uma abordagem conhecida como aprendizagem de representação de grafos [38]. Tais representações permitem a codificação de padrões estruturais em vetores para serem usados por algoritmos de aprendizagem de máquina. Esse é um dos mais novos paradigmas de aprendizagem de máquina e já se mostra promissor em diversas aplicações [39–41]. A outra categoria de algoritmos incluem as redes neurais de grafos [42], que representam abordagens mais recentes e inovadoras. Esses modelos de aprendizado profundo geram representações diferentes, nas quais os vértices agregam iterativamente informações de suas vizinhanças locais para gerar previsões em tarefas de aprendizagem de máquina de ponta a ponta.

Apesar do uso generalizado e dos muitos avanços recentes nos métodos de aprendizagem de máquina para grafos, as aplicações envolvendo redes criminosas são surpreendentemente escassas. Nesse contexto, o presente trabalho contribui para ampliar os estudos nessa área. Investigações empíricas de redes criminosas são muitas vezes dificultadas pela indisponibilidade de dados confiáveis sobre esses sistemas, especialmente dados com aspecto temporal. Em parte, isso ocorre porque criminosos fazem o possível para não serem detectados, mas também porque essas informações são geralmente sigilosas e restritas às agências de aplicação da lei. Nossa trabalho, no entanto, lida com quatro conjuntos de dados de alta qualidade sobre redes criminosas. Duas dessas redes são relacionadas à cor-

rupção, uma ligada a escândalos brasileiros e outra a escândalos espanhóis. Além disso, nossa pesquisa envolve uma parceria com a Polícia Federal Brasileira, que nos forneceu dados inéditos sobre outras duas redes: uma composta por registros relacionados a crimes federais, e outra contendo indivíduos envolvidos em atividades de lavagem de dinheiro.

Nesta tese, apresentamos uma abordagem de ciência de redes e aprendizagem de máquina para investigar redes criminosas e realizar previsões de suas propriedades [43–45]. Esse trabalho está organizado da seguinte maneira. No Capítulo 1 estudamos diversas propriedades das redes de corrupção brasileira e espanhola [43]. O Capítulo 2 apresenta uma aplicação de aprendizagem de máquina em redes criminosas, contendo tarefas relacionadas a previsão de ligações e tipo de associações, além de uma tarefa de regressão para prever a quantidade de dinheiro trocado entre agentes da rede financeira [44]. Em seguida, no Capítulo 3, passamos a considerar aprendizagem profunda para fazer as previsões das mesmas variáveis presentes no Capítulo 2 [45]. Além dessas tarefas, também usamos o algoritmo de redes neurais em grafos para antecipar a reincidência de criminosos. Encerramos essa tese com um resumo das principais contribuições e com nossas perspectivas para pesquisas futuras. Os Apêndices fornecem informações sobre as técnicas e métodos adotados em nossas investigações. O Apêndice A oferece detalhes sobre os métodos de ciência de redes empregados. O Apêndice B aborda conceitos de estatística relevantes para o estudo. Por fim, o Apêndice C apresenta detalhes sobre os métodos de aprendizagem de máquina utilizados em nossa pesquisa.

Capítulo 1

Universalidade de redes de corrupção política

Neste Capítulo, realizamos uma investigação dos aspectos estáticos e dinâmicos de duas redes de corrupção distintas, uma contendo escândalos brasileiros e a outra relacionada a escândalos espanhóis [43]. Inicialmente, apresentamos os conjuntos de dados utilizados e analisamos a distribuição do número de pessoas envolvidas nos casos de corrupção. Em seguida, introduzimos uma abordagem de ciência de redes e construímos as redes de corrupção do Brasil e da Espanha. De posse dessas redes, investigamos suas características e identificamos possíveis estruturas de comunidades.

Após lidar com o cenário estático, tratamos cada rede de corrupção como um sistema dinâmico e estudamos sua evolução. Utilizamos a distribuição de grau como uma medida para caracterizar a topologia dessas redes e analisamos o comportamento dessas distribuições ao longo do tempo. Além disso, também examinamos a relação entre o número de módulos da rede e o seu respectivo número de escândalos. Ainda no contexto dinâmico, investigamos a evolução temporal das redes de corrupção considerando o crescimento de suas maiores componentes, destacando o papel dos agentes reincidentes nesse processo.

Motivados por nossos resultados, propomos um modelo computacional para o crescimento de redes de corrupção baseado na dinâmica de envolvidos reincidentes. Variamos a taxa de reincidência das redes simuladas e estudamos como esse parâmetro influencia suas estruturas. Também realizamos comparações quantitativas e qualitativas entre as redes simuladas e as redes de corrupção empíricas.

Por fim, avaliamos a robustez de nossos resultados em relação ao tamanho dos conjuntos de dados. Dessa forma, verificamos se as propriedades e a dinâmica das redes de corrupção estudadas se mantêm similares mesmo em um cenário em que parte dos dados está faltando.

1.1 Bases de dados

Utilizamos dois conjuntos de dados sobre escândalos de corrupção. O primeiro conjunto de dados contém informações reportadas por Ribeiro *et al.* [19] sobre escândalos de corrupção brasileiros, compreendendo 65 casos bem documentados ocorridos entre 1987 e 2014. Esses dados foram compilados manualmente de páginas da Internet de revistas e jornais de grande circulação e incluem os nomes de 404 pessoas envolvidas.

O segundo conjunto de dados foi compilado ao longo do presente trabalho e contém informações extraídas de um site sem fins lucrativos [46] cujo objetivo é listar todos os escândalos de corrupção espanhóis conhecidos. As informações contidas nesse site também foram extraídas de páginas de revistas de notícias e jornais diários populares na Espanha. Essas informações abrangem 437 escândalos de corrupção que ocorreram entre 1989 e 2018, totalizando 2753 pessoas envolvidas.

Em ambos os conjuntos de dados, realizamos um processamento detalhado para remover informações irrelevantes e dados inválidos ou desatualizados. Além disso, também fizemos uma verificação das informações sobre os principais escândalos de corrupção, comparando-as com outros meios de comunicação.

É importante salientar que a simples menção de uma pessoa em um escândalo de corrupção não garante que ela seja considerada culpada e condenada judicialmente. Processos judiciais relacionados a grandes escândalos de corrupção costumam ser demorados e podem nunca chegar a um veredito final. Por questões legais, anonimizamos os nomes de todos os envolvidos presentes em nossas análises. Ademais, é provável que alguns envolvidos em escândalos de corrupção não tenham sido descobertos e identificados durante as investigações e, nesse sentido, nossos dados podem estar incompletos. No entanto, essa imprecisão é inerente a esse tipo de informação. Uma vez que o foco de nosso estudo é investigar padrões gerais dos processos de corrupção, esses problemas não devem impactar nossos resultados.

Como mostraremos, nossos resultados indicam que diversos padrões relacionados à corrupção e ao crime organizado não dependem de pequenas especificidades dos dados nem das particularidades de cada país. Também mostraremos que os padrões investigados são robustos e estáveis mesmo ao remover 40% de cada conjunto de dados.

1.2 Análise quantitativa dos escândalos de corrupção

Iniciamos nosso estudo tratando do número típico de pessoas por escândalo de corrupção. Conforme relatado na referência [19], a distribuição acumulada complementar do tamanho dos escândalos brasileiros (no quesito número de envolvidos) pode ser ajustada por uma distribuição exponencial¹, com o número típico de pessoas aproximadamente

¹O Apêndice B.1 apresenta algumas propriedades da distribuição exponencial.

igual a 7. A Figura 1.1 mostra que um ajuste exponencial também pode ser aplicado à distribuição acumulada do tamanho dos escândalos espanhóis. Dessa forma, observamos que o número característico de pessoas envolvidas em escândalos de corrupção é de cerca de sete pessoas em ambos os países². É importante notar que ambas as curvas exponenciais subestimam o número de envolvidos nos grandes casos de corrupção. Ainda assim, o modelo exponencial descreve razoavelmente bem uma grande parcela dos dados, uma vez que apenas 20% dos escândalos espanhóis e 17% dos escândalos brasileiros possuem mais de 10 indivíduos.

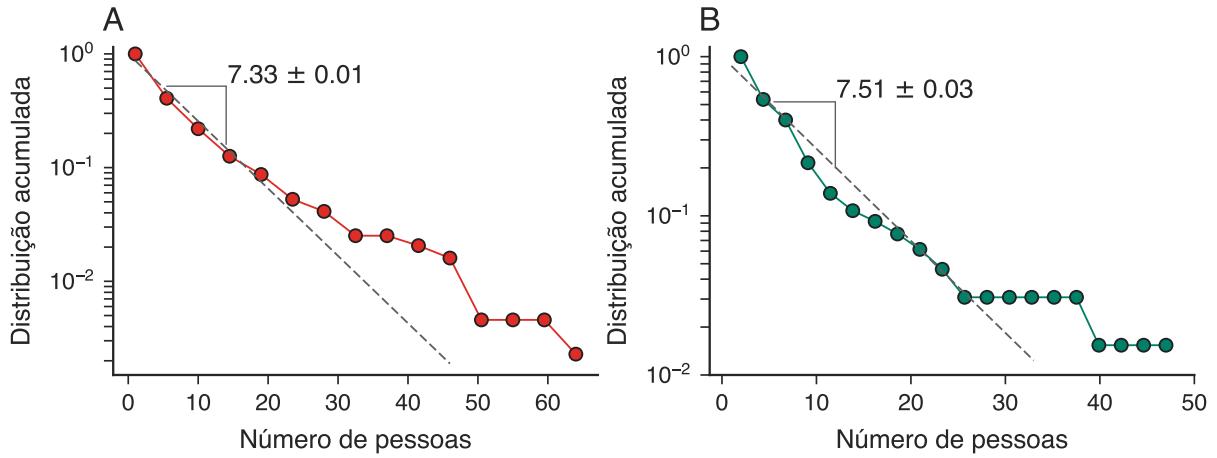


Figura 1.1: Distribuição acumulada complementar do número de implicados em casos de corrupção da (A) Espanha e (B) do Brasil. As linhas tracejadas representam distribuições exponenciais ajustadas aos dados via método de máxima verossimilhança. Esses ajustes indicam que o número típico de envolvidos em casos de corrupção é de aproximadamente 7 pessoas.

O resultado da Figura 1.1 sugere que os envolvidos em corrupção geralmente contam com um pequeno número de parceiros para executar suas atividades criminosas, provavelmente porque tarefas em grande escala são difíceis de gerenciar e permanecerem sigilosas por mais tempo. No geral, as semelhanças encontradas nas distribuições do número de implicados dos dois países indicam um possível padrão universal relacionado aos processos de corrupção política.

1.3 Redes de corrupção política

As atividades de corrupção e crime organizado requerem um alto grau de colaboração entre os envolvidos. Esse aspecto faz com que os sistemas tenham um caráter complexo e comportamentos emergentes difíceis de estudar por meio de abordagens tradicionais de estatística. Para estudar esses tipos de dados é necessária uma abordagem mais com-

²O valor característico em cada ajuste foi obtido por meio do método de máxima verossimilhança. Apresentamos esse método no Apêndice B.2.

pleta. Como já mencionamos, ao longo desse trabalho, adotamos a ciência de redes como principal abordagem para descrever associações criminosas [47, 48].

Em sua forma mais simples, uma rede (ou grafo) é composta por um conjunto de vértices e um conjunto de ligações³. Grafos constituem uma abordagem natural para representar redes de corrupção porque consideram as interações entre todos os envolvidos e, dessa forma, permitem estudar aspectos topológicos desses sistemas. Nossos conjuntos de dados possibilitam criar redes complexas nas quais os vértices representam pessoas envolvidas em corrupção e as conexões indicam que duas pessoas estão implicadas no mesmo escândalo. Portanto, para investigar padrões emergentes de envolvidos em casos de corrupção, construímos as redes de corrupção do Brasil e da Espanha.

Inicialmente, consideramos todos os escândalos de corrupção e ignoramos o aspecto temporal dos dados. A Figura 1.2A mostra a rede de corrupção espanhola considerando todos os 437 escândalos ocorridos entre 1989 e 2018. Essa rede possui 2753 vértices e 27545 ligações, 197 componentes conectadas e 58 vértices isolados. Por outro lado, a Figura 1.2B mostra a rede de corrupção brasileira considerando todos os 65 escândalos ocorridos entre 1987 e 2014. Essa rede é composta por 404 vértices e 3549 ligações formando 14 componentes conectadas.

Apesar de possuírem tamanhos diferentes, essas redes compartilham diversas semelhanças. Quanto aos aspectos estruturais, as redes possuem coeficientes de agrupamento altos e coeficientes de assortatividade relativamente altos, além de densidades baixas e menores comprimentos de caminhos médios⁴. A Tabela 1.1 apresenta os valores das medidas mencionadas.

	Espanha	Brasil	Espanha (maior componente)	Brasil (maior componente)
Coeficiente de agrupamento	0.91	0.93	0.94	0.93
Coeficiente de assortatividade	0.74	0.53	0.59	0.50
Densidade	0.007	0.044	0.025	0.06
Comprimento médio do caminho			5.11	2.99

Tabela 1.1: Valores de medidas estruturais das redes de corrupção brasileira e espanhola.

Além dos aspectos anteriores, procuramos identificar estruturas de comunidades (ou módulos) nas redes de corrupção. Essa é uma propriedade encontrada frequentemente em redes empíricas, nas quais existem vértices mais densamente conectados entre si do que com outros grupos de vértices da rede [49–51]. Calderoni *et al.* [52] afirmam que redes criminosas tendem a formar comunidades para reduzir o risco de vazamento de informações. A detecção dessas comunidades em redes criminosas fornece informações que vão além de revelar sua estrutura interna e podem auxiliar, por exemplo, na identificação de líderes e

³O Apêndice A contém os principais conceitos de ciência de redes utilizados neste trabalho.

⁴O Apêndice A.3 apresenta a definição das medidas estruturais mencionadas.

vértices importantes na rede [52] ou até mesmo encontrar vértices que fragmentam a rede de forma mais eficiente [23, 53].

Ribeiro *et al.* [19] identificaram 27 módulos na rede de corrupção brasileira. Esse resultado mostra que o número de escândalos (65) é aproximadamente duas vezes maior que o número de módulos, indicando que existem vários escândalos que poderiam ser agrupados quanto a sua composição. Tendo em vista esse resultado, aplicamos o algoritmo *Infomap*⁵ para detectar comunidades nas redes espanhola e brasileira. Com isso, encontramos que a rede de corrupção espanhola também possui uma estrutura modular e, similarmente à rede brasileira, esses módulos tentam a mesclar mais de um escândalo. De fato, a razão entre o número de módulos e o número de escândalos é de 0.76 para a rede espanhola e 0.62 para a rede brasileira. As cores presentes nas redes das Figuras 1.2A e 1.2B referem-se a cada módulo encontrado, de tal forma que módulos de tamanhos próximos possuem cores parecidas.

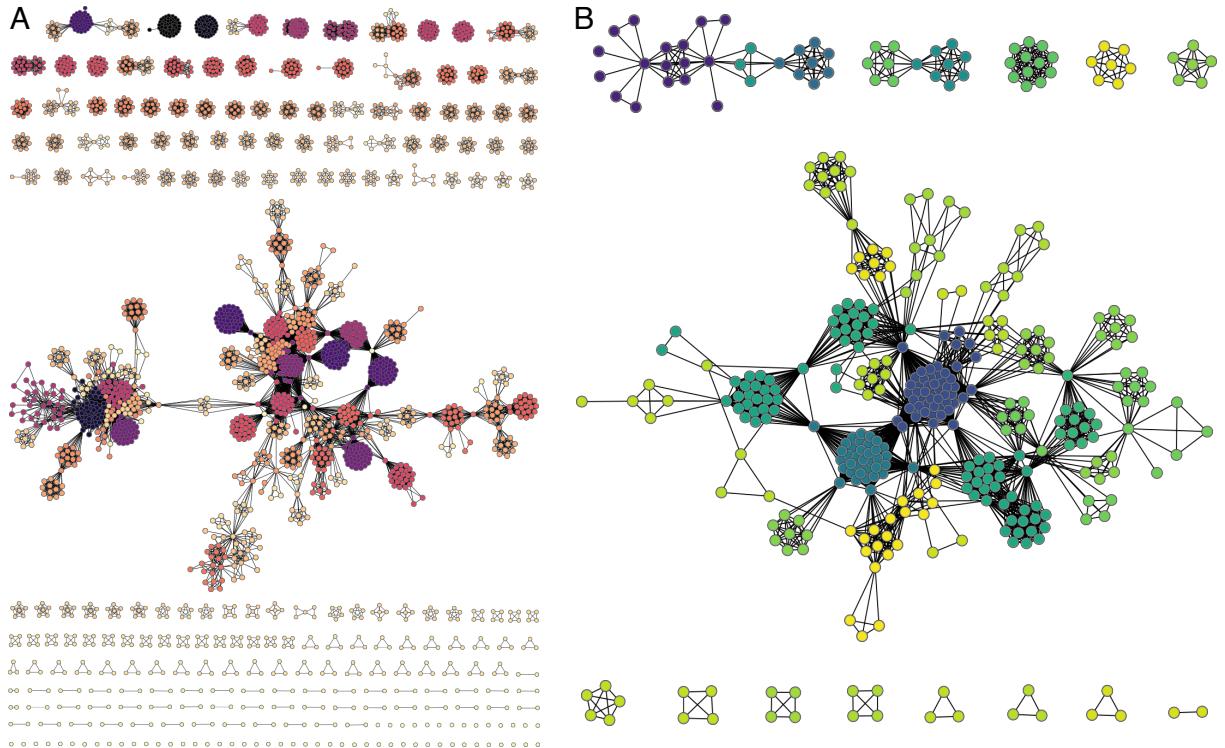


Figura 1.2: Visualização das redes de corrupção formadas por pessoas envolvidas em escândalos políticos da (A) Espanha e do (B) Brasil. Em ambas as redes, os vértices representam pessoas e as ligações entre eles indicam indivíduos envolvidos em ao menos um mesmo caso de corrupção. As cores referem-se às estruturas modulares dessas redes estimadas pelo algoritmo *Infomap*.

⁵O Apêndice A.4 ilustra esse algoritmo. Embora não exista um método à prova de falhas para detecção de comunidades ou estrutura modular em redes, utilizamos o algoritmo *Infomap* devido a sua eficiência computacional. Ademais, encontramos resultados semelhantes com maximização de modularidade [51] e modelos de blocos estocásticos [54].

1.4 Evolução temporal da distribuição de grau

Até agora, nos concentramos exclusivamente nos estágios finais das redes de corrupção. As estruturas mostradas nas Figuras 1.2A e 1.2B são resultantes do crescimento acumulado ao longo de todos os anos dessas redes. É importante destacar que essas estruturas apresentam vários estágios, dependendo do ano limite considerado para os escândalos. Mais especificamente, as redes crescem devido ao surgimento de novos escândalos e de agentes reincidientes (pessoas envolvidas em mais de um caso de corrupção). A partir desta seção, abordamos as redes de corrupção como sistemas dinâmicos e estudamos suas características em cada um dos seus estágios. Nossa objetivo é investigar os mecanismos responsáveis pela formação e evolução das propriedades dessas redes.

Uma das maneiras mais diretas para caracterizar a topologia de uma rede complexa é por meio de sua distribuição de grau⁶. Conforme reportado por Ribeiro *et al.* [19], a distribuição de grau da rede de corrupção brasileira pode ser aproximada por uma distribuição exponencial com valor característico de aproximadamente 17 conexões.

Visando verificar se a rede de corrupção espanhola também apresenta esse comportamento, estimamos sua distribuição de grau e comparamos os resultados com a distribuição da rede de corrupção brasileira. Uma vez que nossos dados permitem investigar padrões dinâmicos associados ao crescimento dessas redes, estimamos as distribuições de grau ao longo do tempo. Nossa intuito é descobrir se a distribuição de grau de todos os anos apresenta uma tendência exponencial.

Em escala mono-logarítmica, os painéis (A) e (B) da Figura 1.3 contêm, respectivamente, as distribuições de grau da rede de corrupção espanhola e da rede de corrupção brasileira. As inserções nesses painéis mostram as distribuições de grau para o estágio mais recente de cada rede. Observamos um comportamento aproximadamente linear dessas distribuições, indicando que a distribuição exponencial representa uma boa primeira aproximação para as distribuições de grau. Portanto, aplicamos o método de máxima verossimilhança aos dados, ajustamos o modelo exponencial e obtemos uma estimativa do grau característico dessas redes (20 para a rede espanhola e 17.6 para a rede brasileira, considerando apenas o estágio mais recente das redes).

Considerando o aspecto temporal, as curvas das Figuras 1.3A e 1.3B representam as distribuições de grau das redes ao longo dos anos. Mais especificamente, calculamos as distribuições de grau para cada ano da rede após redimensionar o grau dos vértices pelo valor médio do grau nos anos correspondentes. Ou seja, para cada ano, os graus dos vértices são divididos por seu respectivo valor médio. Considerando a hipótese exponencial⁷, essa operação de reescala deve fazer as distribuições de grau dos diferentes anos colapsarem em uma única distribuição com grau característico unitário.

⁶O Apêndice A.2.2 apresenta detalhes sobre distribuições de grau.

⁷Informações sobre a distribuição exponencial (e hipótese exponencial) podem ser encontradas no Apêndice B.1.

As Figuras 1.3A e 1.3B revelam que existe uma tendência das distribuições de grau se aproximarem da distribuição com valor característico unitário, indicando que a distribuição exponencial é uma boa aproximação para os dados. Para investigar o comportamento do grau característico, calculamos esse valor para cada distribuição ao longo dos anos. Sob a hipótese exponencial, é importante saber se o grau característico exibe alguma tendência. As Figuras 1.3C e 1.3D mostram, respectivamente, a evolução do grau característico das distribuições da rede espanhola e da rede brasileira. Nessas figuras, as barras de erro representam um intervalo de confiança de 95% obtido por *bootstrap*⁸.

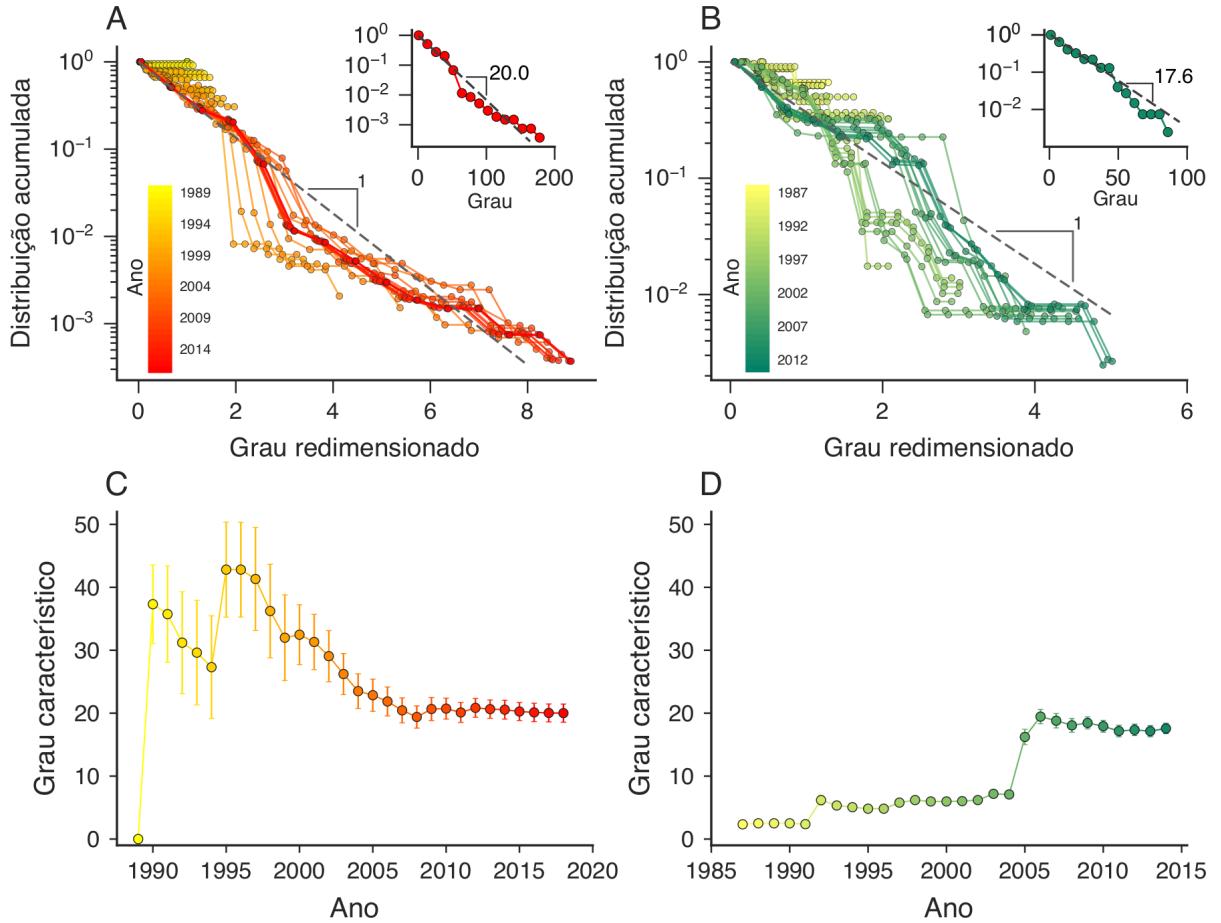


Figura 1.3: Distribuições acumuladas complementares de grau das redes de corrupção (A) espanhola e (B) brasileira. As curvas representam distribuições de grau para cada ano da rede após reescalar o grau dos vértices pelo valor médio do grau nos anos correspondentes. As inserções mostram as distribuições de graus para o estágio mais recente da rede de cada país. As cores das curvas correspondem aos diferentes anos das redes, conforme o código de cores presentes nas barras à esquerda. Evolução do grau característico das redes de corrupção (C) espanhola e (D) brasileira. As barras de erro indicam intervalos de confiança de 95% estimados por *bootstrap*.

Em ambas as curvas, observamos variações significativas até 2006, seguidas por um grau característico aproximadamente estável nos anos posteriores. É interessante observar

⁸Descrevemos o procedimento de *bootstrap* no Apêndice B.3.

que o grau característico da rede espanhola estabiliza em um valor próximo ao grau característico da rede de corrupção brasileira. Essa semelhança reflete um comportamento possivelmente universal de redes de corrupção, isto é, apesar das especificidades de cada país e de seus escândalos de corrupção, a forma exponencial da distribuição de grau e o seu grau característico não mudam muito entre os dois sistemas. Em outras palavras, os padrões de conexão nessas redes parecem ser independentes de aspectos específicos de cada país.

1.5 Dinâmica de crescimento das redes de corrupção

As redes de corrupção evoluem no tempo devido ao surgimento e descoberta de novos escândalos de corrupção. Para tentar encontrar padrões nessa evolução temporal, uma abordagem natural é considerar o crescimento das maiores componentes das redes. Em teoria dos grafos, uma componente conectada constitui um subgrafo da rede. Subgrafos representam conjuntos disjuntos e, portanto, não há conexão entre eles. No entanto, dentro de cada subgrafo existe pelo menos um caminho entre qualquer par de vértices.

Em muitos casos, redes complexas possuem componentes de diversos tamanhos. Em geral, estamos interessados na dinâmica das maiores componentes da rede. Mais especificamente, estudamos a evolução temporal de uma componente usando como medida o seu número de vértices. Considerando o último estágio (ano) dos nossos dados, a rede de corrupção espanhola possui uma componente gigante que representa 40% de todos os vértices e 53% de todas as ligações da rede. Por outro lado, a rede de corrupção brasileira contém uma componente gigante com 77% dos vértices e 93% das ligações da rede. Portanto, essas componentes representam frações significativas desses sistemas.

O processo que leva redes complexas a possuírem uma componente gigante se assemelha com uma transição de fase [47], no qual componentes da rede se agrupam para criar uma componente gigante. De fato, no caso da rede de corrupção brasileira, Ribeiro *et al.* [19] mostram que a evolução da rede ocorre por meio de processos de coalescência de componentes isoladas, nas quais existe um número muito reduzido de indivíduos responsáveis pela interligação de diferentes grupos de escândalos de corrupção.

A Figura 1.4 apresenta a evolução temporal do tamanho das duas maiores componentes das redes de corrupção (A) espanhola e (B) brasileira. Observamos que o crescimento de ambas as redes exibe um fenômeno de coalescência. Em particular, na Figura 1.4A, existe um crescimento abrupto da maior componente de 2011 para 2012. Ao mesmo tempo, também notamos que a segunda maior componente diminui em tamanho. Por outro lado, as curvas da Figura 1.4B apresentam um comportamento similar entre os anos 2004 e 2005, com a maior componente aumentando de tamanho abruptamente e a segunda maior componente diminuindo de tamanho concomitantemente. Uma vez que envolvidos nunca são removidos da rede, esse comportamento indica que, em ambas as redes, as maiores

componentes se juntam formando uma só.

Com o objetivo de investigar esse comportamento de forma mais qualitativa, identificamos as maiores componentes presentes nas redes durante os anos de variação abrupta. As Figuras 1.4C e 1.4D identificam, respectivamente, as maiores componentes da rede de corrupção espanhola e da rede de corrupção brasileira. Notamos que a rede de corrupção espanhola passa por uma transição que resulta da coalescência das três maiores componentes (mostradas nas cores laranja, amarela e vermelha) presentes no ano de 2011, formando a componente gigante de 2012. Já a rede de corrupção brasileira apresenta uma junção de componentes de forma similar, mas apenas com as duas maiores componentes (mostradas nas cores verde-claro e verde-escuro). Nesses painéis, vértices coloridos de cinza representam novos implicados não envolvidos em escândalos anteriores.

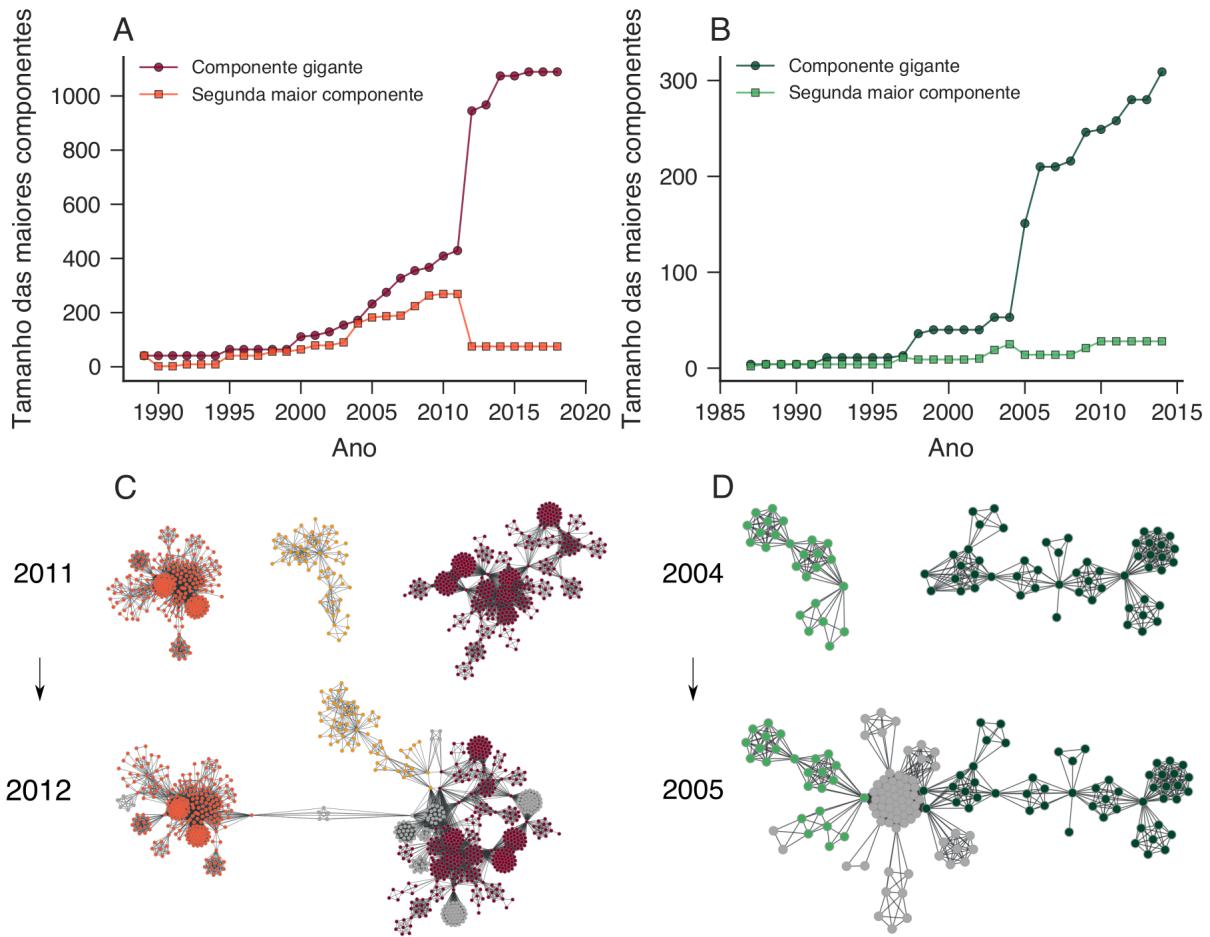


Figura 1.4: Evolução temporal do tamanho das maiores componentes das redes de corrupção (A) espanhola e (B) brasileira. Processo de coalescência entre as maiores componentes das redes de corrupção (C) espanhola e (D) brasileira. Nessa figura, as cores vermelho, laranja e amarela representam, respectivamente, a primeira, segunda e terceira maior componente da rede de corrupção espanhola. Por outro lado, as cores verde-escuro e verde-claro representam, respectivamente, a primeira e segunda maior componente da rede de corrupção brasileira. Os demais envolvidos implicados em 2012 [painel (C)] e em 2005 [painel (D)] estão coloridos de cinza.

No geral, nossos resultados indicam que poucos envolvidos reincidentes são responsáveis pela formação das componentes gigantes dessas redes. Mais precisamente, envolvidos presentes em 2011 (rede de corrupção espanhola) ou em 2004 (rede de corrupção brasileira) reincidem nos anos seguintes em novos escândalos de corrupção. Esses escândalos, por sua vez, fazem o papel de “ponte” entre diferentes componentes, produzindo o efeito de coalescência. A partir do comportamento observado, concluímos que existe uma semelhança na dinâmica de crescimento das redes de corrupção brasileira e espanhola. Independentemente das razões e especificidades desse processo em cada país, essa dinâmica depende de um pequeno número de envolvidos reincidentes. Dessa forma, nossos achados revelam outro possível aspecto universal presente na evolução de redes de corrupção.

1.6 Comportamentos lineares nas redes de corrupção

Conforme relatamos, o estágio mais recente das redes de corrupção apresenta uma estrutura de comunidades na qual dois ou mais escândalos tendem a se mesclar em um único módulo. Nesse contexto, queremos saber se esse comportamento é específico do último estágio das redes ou uma propriedade mais geral que persiste no tempo. Para responder a essa pergunta, para cada rede calculamos a relação entre o número acumulado de módulos e o respectivo número acumulado de escândalos de corrupção ao longo do tempo. As Figuras 1.5A e 1.5B mostram essa relação.

Observamos que, em ambos os países, o número de módulos das redes cresce linearmente com o total de escândalos políticos. Ajustamos um modelo linear aos dados de cada país e encontramos um bom acordo com o modelo. Obtemos um coeficiente angular de 0.744 módulos por escândalo para a Espanha (Figura 1.5A) e 0.626 módulos por escândalo para o Brasil (Figura 1.5B). Portanto, apesar da complexidade subjacente aos processos de corrupção, a estrutura das redes de corrupção preserva, aproximadamente, a relação entre número de módulos e escândalos ao longo de todo o seu processo de crescimento. Vale ressaltar que uma condição para o equilíbrio entre o número de módulos e escândalos é o surgimento de agentes reincidentes responsáveis por conectar diferentes escândalos políticos.

Em linhas gerais, a dinâmica das maiores componentes da rede e a associação linear entre módulos e escândalos expõem o papel crítico dos agentes reincidentes na estrutura das redes de corrupção. Sendo assim, voltamos nossa atenção ao papel dos envolvidos reincidentes em atividades de corrupção. Primeiramente, lembramos que esses vértices fazem a conexão entre dois ou mais escândalos e, portanto, são os principais responsáveis pela estrutura das redes de corrupção do Brasil e da Espanha. Nossa estudo também revela que o processo tipo coalescência, observado nas redes de corrupção espanhola (Figura 1.4A) e brasileira (Figura 1.4B), depende de um pequeno número de envolvidos reincidentes.

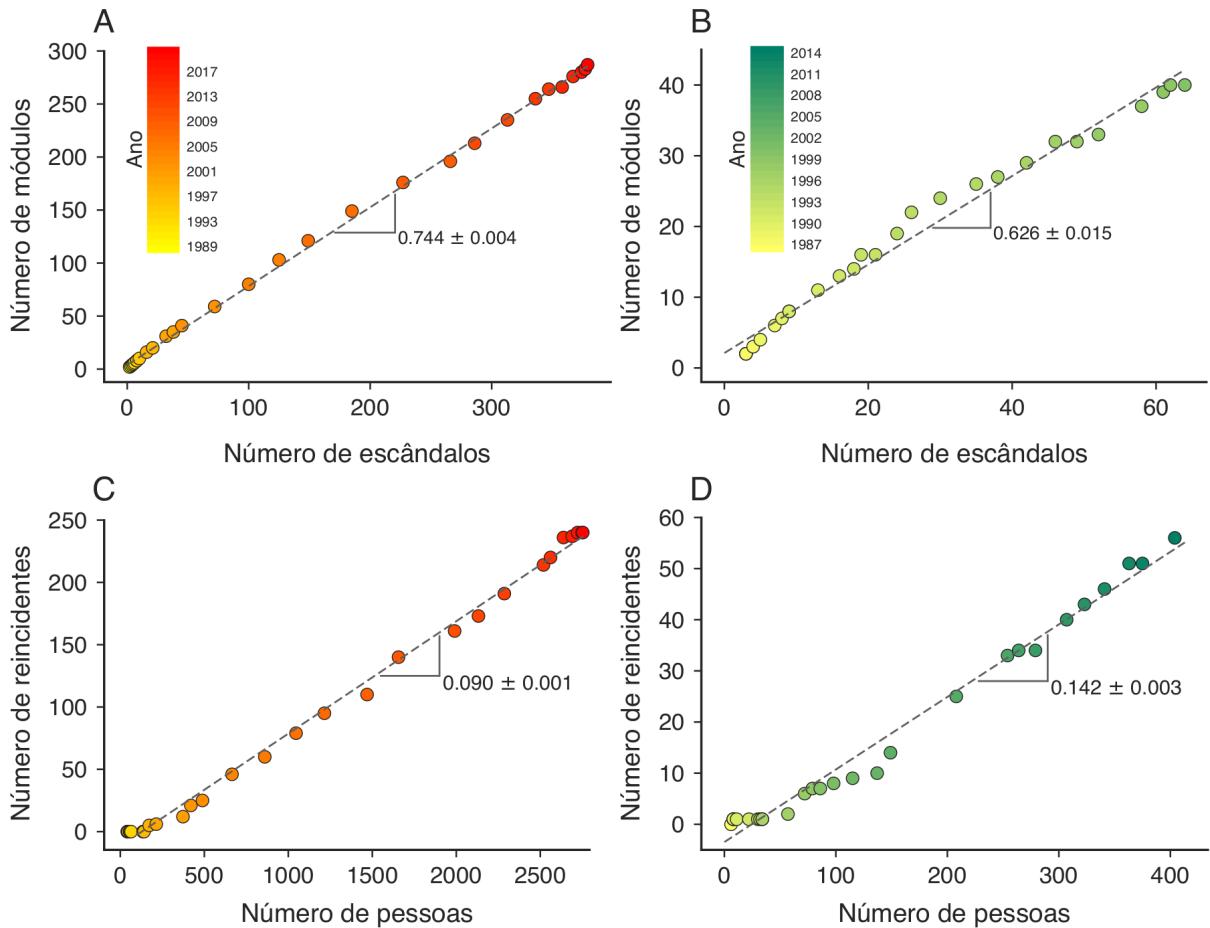


Figura 1.5: Relação entre o número de módulos e de escândalos para cada ano das redes de corrupção (A) espanhola e (B) brasileira. As linhas tracejadas representam modelos lineares ajustados aos dados, com uma taxa de 0.744 ± 0.004 módulos por escândalo na rede espanhola e 0.626 ± 0.015 módulos por escândalo na rede brasileira. Associação entre o número de agentes reincidentes e o número total de pessoas para cada ano das redes de corrupção (C) espanhola e (D) brasileira. As linhas tracejadas representam modelos lineares ajustados aos dados, com 0.090 ± 0.001 e 0.142 ± 0.003 reincidentes por pessoa nas redes espanhola e brasileira, respectivamente. Os valores dos coeficientes lineares (interceptos dos modelos) obtidos por esses ajustes são, respectivamente, -11.5 ± 1.599 e -3.468 ± 0.684 .

Para entender melhor o surgimento desses vértices especiais, investigamos como o número de envolvidos reincidentes aumenta à medida que novos escândalos são descobertos e adicionados às redes de corrupção. As Figuras 1.5C e 1.5D mostram a relação entre o número acumulado de agentes reincidentes e o total de pessoas em cada ano nas redes de corrupção. Notamos que essas duas quantidades estão linearmente associadas, implicando que os agentes se tornam reincidentes a uma taxa aproximadamente constante ao longo dos anos. Ajustando um modelo linear a esses dados, encontramos uma taxa de reincidência de 0.090 ± 0.001 reincidentes por pessoa para a Espanha e de 0.142 ± 0.003 reincidentes por pessoa para o Brasil. Essas taxas indicam que esperamos encontrar cerca de 9 reincidentes a cada 100 agentes corruptos na rede espanhola. Em comparação, a rede brasileira possui

em torno de 14 reincidentes a cada 100 agentes corruptos.

Infratores reincidentes são os elos entre escândalos de corrupção, conectando diferentes partes da rede. Nesse sentido, a taxa de reincidência criminosa desempenha um papel importante na estrutura e dinâmica das redes de corrupção. De fato, a diferença entre as taxas de reincidência do Brasil e da Espanha pode explicar parcialmente a diferença entre as medidas estruturais de suas redes. Por exemplo, uma maior taxa de reincidência criminosa torna a rede mais densa. A componente gigante da rede de corrupção brasileira possui um valor de densidade igual a 0.0689, quase três vezes o valor da densidade da maior componente da rede espanhola (0.0245). Além disso, um valor maior dessa taxa também torna a rede mais compacta. A componente gigante da rede de corrupção brasileira possui menor comprimento de caminho médio (2.99 versus 5.11) e menor diâmetro (7 versus 11) do que a maior componente da rede espanhola.

1.7 Modelo para redes de corrupção

Motivados por nossas descobertas empíricas e pelas semelhanças entre as redes espanhola e brasileira, propomos um modelo computacional para o crescimento de redes de corrupção que se baseia na dinâmica de envolvidos reincidentes. Por uma perspectiva estrutural, envolvidos reincidentes formam o “esqueleto” das redes de corrupção, sendo os principais responsáveis pelo crescimento e estrutura macroscópica desses sistemas. Os demais vértices complementam as redes e também são importantes, sobretudo em termos de resiliência, uma vez que a remoção desses agentes causa menos impacto na estrutura desses sistemas.

Nosso modelo pode ser descrito da seguinte maneira. A rede começa vazia e cresce pela inclusão de grafos completos a cada iteração. Um grafo completo representa um caso de corrupção porque todos os seus vértices estão conectados, da mesma maneira que pessoas envolvidas em um escândalo também estão todas conectadas. O tamanho s de cada grafo completo (no quesito número de vértices) é sorteado aleatoriamente de uma distribuição exponencial $P(s)$, a qual imita o comportamento empírico da Figura 1.1. Mais especificamente, escolhemos $P(s) \sim e^{-s/s_c}$, na qual s_c é o tamanho característico dos escândalos de corrupção que, empiricamente, é de aproximadamente sete pessoas.

Durante o crescimento da rede, consideramos que parte dos vértices adicionados à rede a cada iteração atuam como reincidentes. Seguindo o comportamento empírico das Figuras 1.5C e 1.5D, assumimos que o número de reincidentes (r) aumenta linearmente com o número total de agentes (n) via $r(n) = \alpha n + \beta$, em que α é a taxa de reincidência e β é um coeficiente negativo que controla o número mínimo de pessoas necessário para o surgimento dos primeiros agentes reincidentes. Acompanhamos o número de reincidentes durante o processo de crescimento da rede e, quando esse número aumenta, selecionamos aleatoriamente vértices já presentes na rede para se tornarem reincidentes e fazê-los

pertencer ao próximo escândalo (grafo completo) a ser adicionado.

Na definição atual de nosso modelo, apenas vértices não recorrentes podem reincidir. No entanto, as redes empíricas apresentam envolvidos que reincidem em mais de dois casos de corrupção. Encontramos que, em média, durante o crescimento das redes de corrupção espanhola e brasileira, uma pequena fração (2.5%) de todos os indivíduos no estágio final da rede satisfazem esse critério. Portanto, consideramos esse valor como um parâmetro ($p_a = 0.025$) adicional do modelo. Incluímos esse comportamento em nosso modelo de modo que, ao selecionar um vértice recorrente, existe uma probabilidade p_a desse vértice já ter reincidido anteriormente. Em nossas análises, observamos que esse parâmetro afeta pouco a estrutura da rede, sobretudo quando $p_a < 0.1$.

O aspecto principal desse modelo é a taxa de envolvidos reincidentes α . Esse parâmetro controla a proporção de envolvidos reincidentes na rede e, consequentemente, seu valor altera substancialmente a estrutura da rede gerada. Portanto, o estudo do modelo com diferentes valores de α pode fornecer informações importantes, revelando o papel desempenhado por esse tipo de vértice na estrutura das redes formadas pelo modelo. De forma mais específica, nossa investigação sobre o comportamento do modelo será realizada estudando o tamanho da maior componente das redes geradas para vários valores de α .

Em nossa análise, fixamos os outros parâmetros do modelo ($p_a = 0.025$, $\beta = -11.5$ e $s_c = 7$) e variamos apenas α . A Figura 1.6 mostra a média do tamanho da maior componente das redes obtidas via nosso modelo em função de α . Para cada valor de α , crescemos a rede por meio de 1000 iterações do modelo e extraímos a fração do tamanho da maior componente da rede gerada pela última iteração. Por fim, repetimos esse processo 1000 vezes e calculamos uma média do tamanho da maior componente para cada um dos valores de α . A banda colorida (em verde marinho claro) representa os valores máximo e mínimo de cada um desses conjuntos de 1000 realizações⁹.

O gráfico inserido no centro da Figura 1.6 corresponde à derivada da curva citada anteriormente. Construímos essa derivada para destacar a variação relativamente brusca ocorrida nessa curva em torno de $\alpha \approx 0.065$. Notamos que o gráfico apresenta um ponto máximo nesse valor e, como veremos adiante, esse comportamento é similar a uma transição de fase das redes formadas pelo modelo.

As visualizações das redes inseridas na Figura 1.6 representam formas típicas das redes geradas para três valores diferentes de α . Observamos que se o valor de α é muito pequeno ($\alpha \rightarrow 0$), a rede formada é esparsa e composta por um grande número de componentes isoladas. No outro extremo, valores de α próximos de 1 produzem redes muito conectadas e apresentam uma estrutura em forma de “corrente”. Entretanto, o modelo apresenta uma espécie de transição de fase entre esses dois extremos ao redor de $\alpha_c \approx 0.065$ (ponto crítico do modelo). Na Figura 1.6, a linha tracejada em cinza indica a taxa de reincidência que

⁹Nesse caso, usamos os valores máximos e mínimos porque os erros padrões das médias são muito pequenos e não ficam visíveis na figura. Isso significa que as médias variam muito pouco.

coincide com o ponto crítico do modelo (α_c). Curiosamente, essa taxa de reincidência crítica é relativamente próxima às taxas empíricas estimadas para as redes espanhola ($\alpha = 0.09$) e brasileira ($\alpha = 0.142$). Portanto, os processos de corrupção parecem operar próximos a uma taxa crítica de reincidência, abaixo da qual a rede se torna totalmente fragmentada e acima da qual ela é excessivamente conectada.

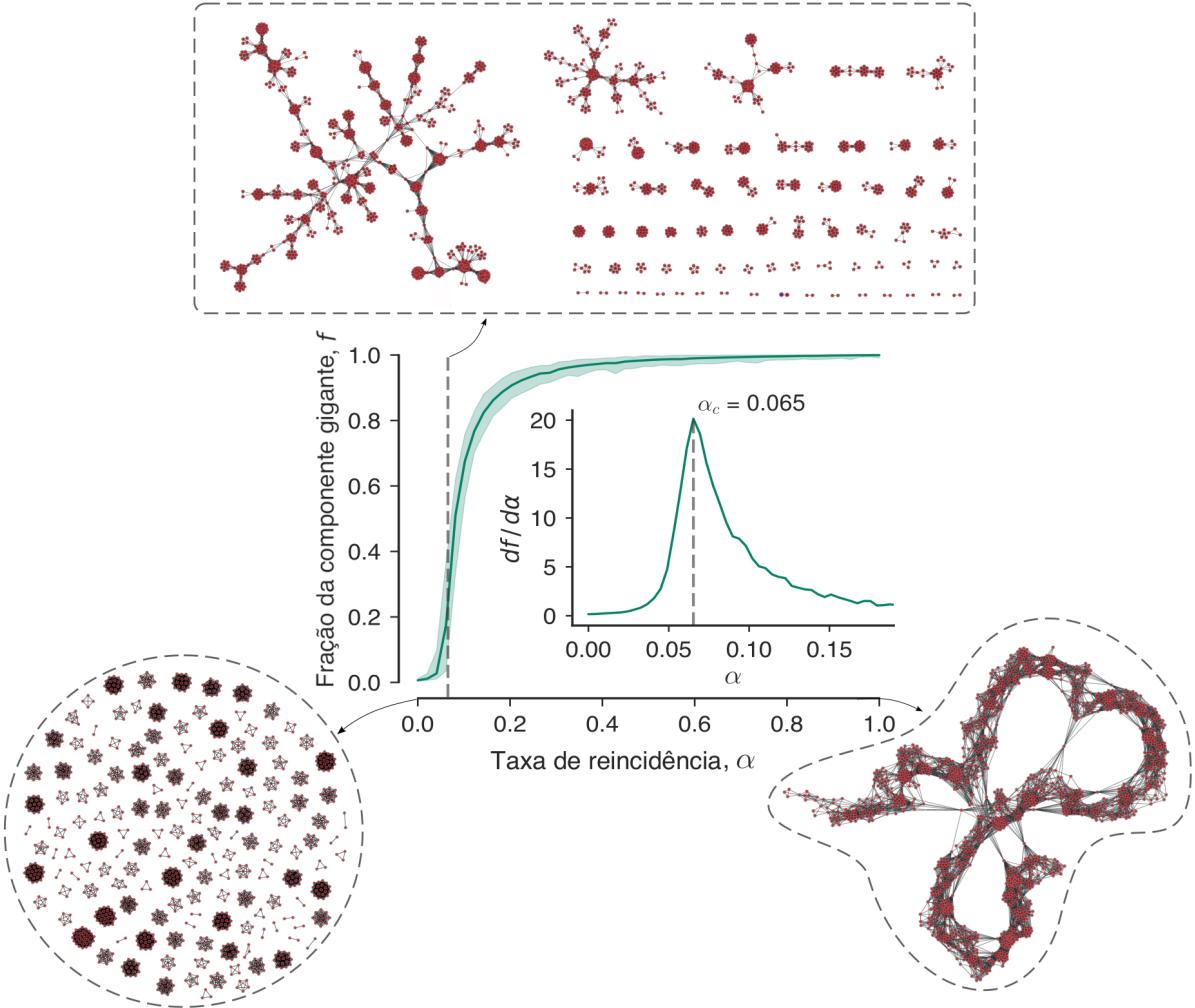


Figura 1.6: Tamanho médio da maior componente das redes obtidas via nosso modelo em função da taxa de reincidência α . A curva inserida internamente nesse gráfico representa a derivada da quantidade anterior. Nessa curva, observamos um valor máximo, correspondendo a uma espécie de transição de fase do modelo. Em ambas as curvas, as linhas verticais tracejadas representam o mesmo ponto ($\alpha = 0.065$) correspondente a essa transição. Os valores empíricos obtidos para as taxas de reincidência das redes de corrupção espanhola ($\alpha = 0.090$) e brasileira ($\alpha = 0.142$) são relativamente próximos ao valor da transição ($\alpha = 0.065$). As três redes mostradas na figura correspondem a formas típicas das redes geradas para $\alpha = 0$, $\alpha = 0.065$ e $\alpha = 1$.

Investigamos também o comportamento do modelo próximo a essa taxa crítica. Essa análise é relevante porque nos permite ter mais confiança que esse valor é estável e que o número de iterações usado anteriormente é suficiente para encontrar o ponto onde ocorre

o máximo da derivada. Em outras palavras, queremos investigar se α_c se desloca caso as redes do modelo sejam maiores. Além disso, essa análise também ajuda a entender o tipo de mudança que existe nesse intervalo (isto é, se essa transição é suave ou brusca).

A Figura 1.7A mostra a fração média da componente gigante das redes simuladas em função da taxa de reincidência (α) ao passo que aumentamos o número máximo de iterações (grafos completos, t). Diferentemente da análise presente na Figura 1.6, onde apresentamos a fração da maior componente das redes crescidas até $t = 1000$ grafos completos, agora variamos t de 100 até 10000 e examinamos o comportamento das curvas geradas. A Figura 1.7B mostra as derivadas ($df/d\alpha$) das curvas anteriores. À medida que t aumenta, as taxas de reincidência que produzem os valores máximos se aproximam de α_c e a curva atinge altura máxima nesse ponto (indicado pela linha vertical tracejada).

A análise do comportamento presente na Figura 1.7 sugere que a transição de redes fragmentadas para redes conectadas é suave. Essa característica indica que a transição de fase ocorrida no modelo é de segunda ordem, uma vez que não existe descontinuidade na primeira derivada. Esse processo é semelhante às transições de fase de modelos de percolação [55].

Após o estudo detalhado do modelo, comparamos o comportamento das redes simuladas com os resultados empíricos. Para fazer isso, crescemos 100 redes usando a taxa de reincidência da Espanha ($\alpha = 0.09$) e outras 100 redes usando a taxa de reincidência do Brasil ($\alpha = 0.142$), novamente fixando os parâmetros $p_a = 0.025$, $\beta = -11.5$ e $s_c = 7$. Nessas simulações, o número de grafos completos adicionados às redes é igual ao número total de escândalos em cada conjunto de dados (437 para a Espanha e 65 para o Brasil).

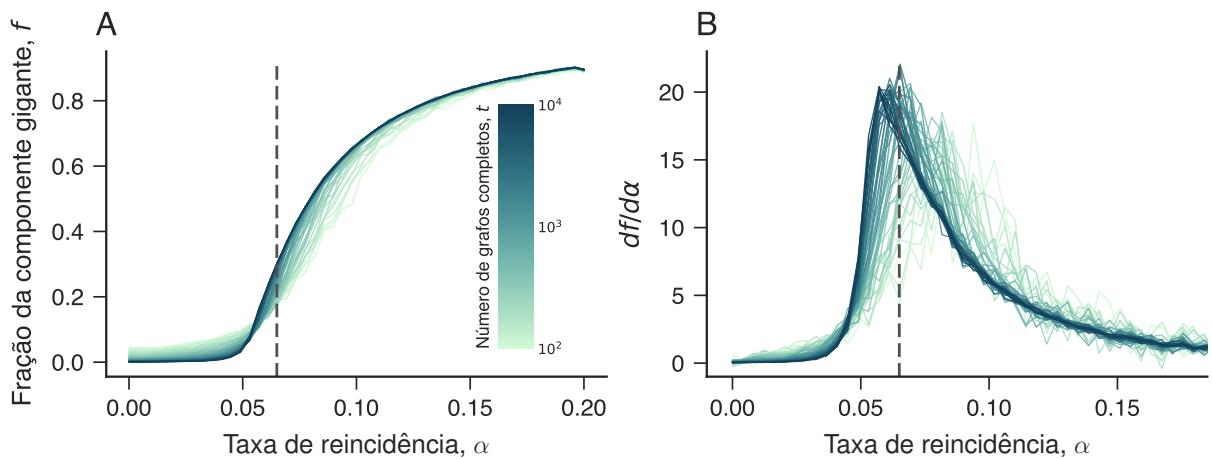


Figura 1.7: (A) Fração média da componente gigante das redes simuladas (f) em função da taxa de reincidência (α) variando o número de grafos completos (t , indicado pelo código de cores). (B) Derivada de f em relação a α . As redes crescidas via nosso modelo possuem os seguintes parâmetros fixos: $s_c = 7$, $\beta = 12$, $p_a = 0.025$. Em ambos os painéis, a linha vertical tracejada indica a taxa de reincidência crítica ($\alpha_c = 0.065$).

Antes de compararmos quantitativamente as redes do modelo com as redes empíri-

cas, observamos qualitativamente que as redes simuladas são visualmente semelhantes às contrapartes empíricas. A Figura 1.8 apresenta essa comparação.

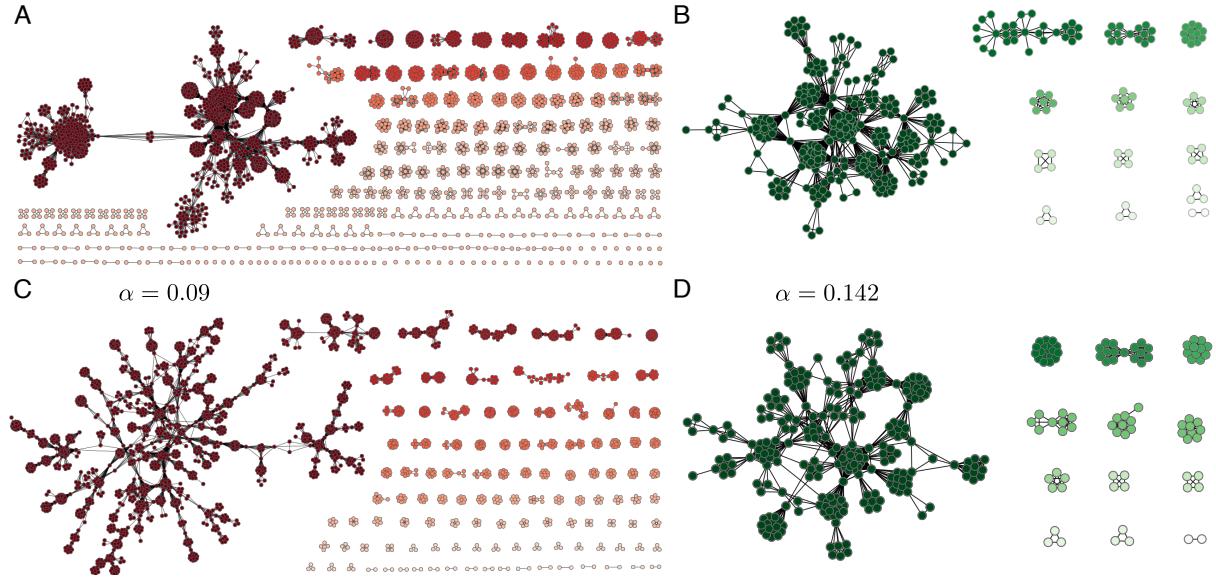


Figura 1.8: Comparação visual entre as redes de corrupção espanhola e brasileira [painéis (A) e (B), respectivamente] e redes típicas geradas por nosso modelo usando os valores empíricos da taxa de reincidência da Espanha [$\alpha = 0.09$, painel (C)] e do Brasil [$\alpha = 0.142$, painel (D)]. Em cada simulação, o número de iterações é igual ao número de escândalos da sua contraparte empírica. As cores dos vértices distinguem as componentes conectadas das redes.

Em seguida, comparamos as redes simuladas com as redes de corrupção por meio das mesmas medidas calculadas anteriormente (coeficiente de agrupamento, coeficiente de assortatividade, densidade e comprimento médio de caminho). A Tabela 1.2 apresenta essa comparação. Em geral, notamos que as redes simuladas produzem medidas estruturais próximas aos valores de suas contrapartes empíricas.

	Espanha (simulação)	Brasil (simulação)	Espanha (simulação, maior componente)	Brasil (simulação, maior componente)
Coeficiente de agrupamento	0.91 (0.949 ± 0.003)	0.93 (0.938 ± 0.008)	0.94 (0.945 ± 0.004)	0.93 (0.938 ± 0.007)
Coeficiente de assortatividade	0.74 (0.76 ± 0.01)	0.53 (0.67 ± 0.03)	0.59 (0.69 ± 0.02)	0.50 (0.63 ± 0.04)
Densidade	0.007 (0.0042 ± 0.0002)	0.044 (0.030 ± 0.003)	0.025 (0.0084 ± 0.0009)	0.06 (0.043 ± 0.008)
Comprimento médio do caminho			5.11 (9.17 ± 1.01)	2.99 (4.87 ± 0.55)

Tabela 1.2: Valores de medidas estruturais das redes empíricas e simuladas. A tabela mostra os valores empíricos acima das medidas calculadas nas redes simuladas. Os números entre parênteses representam o valor médio ± 1 desvio padrão de um conjunto de 100 simulações usando a taxa de reincidência das redes empíricas.

Além das propriedades estáticas, verificamos que nosso modelo também reproduz o

processo de crescimento das redes de corrupção empíricas. Em particular, descobrimos que as distribuições de grau das redes simuladas também são bem descritas por distribuições exponenciais. Os gráficos inseridos nas Figuras 1.9A (com $\alpha = 0.09$) e 1.9B (com $\alpha = 0.142$) mostram essas distribuições de grau para redes geradas com $t = 800$ grafos completos. Nessas figuras, notamos um aspecto linear da distribuição na escala monologarítmica e um bom ajuste com o modelo exponencial.

Nesse mesmo contexto, investigamos a evolução temporal do grau característico dessas distribuições simuladas e comparamos com a evolução do grau característico das redes reais de corrupção. Para isso, simulamos as redes do modelo usando as taxas de reincidência das suas contrapartes empíricas. Crescemos redes por meio de 800 iterações e, em cada passo, calculamos o valor característico da sua distribuição de grau. Repetimos esse processo 100 vezes e calculamos a média desse conjunto. As Figuras 1.9A e 1.9B mostram, respectivamente, a média desse parâmetro com simulações usando as taxas de reincidência $\alpha = 0.09$ e $\alpha = 0.142$ (valores correspondentes das redes espanhola e brasileira). Visualmente, a evolução da distribuição de grau se assemelha à evolução da distribuição de grau das redes empíricas (Figura 1.3), com um grau característico inicialmente variando bastante e depois apresentando um platô para estágios posteriores das redes simuladas.

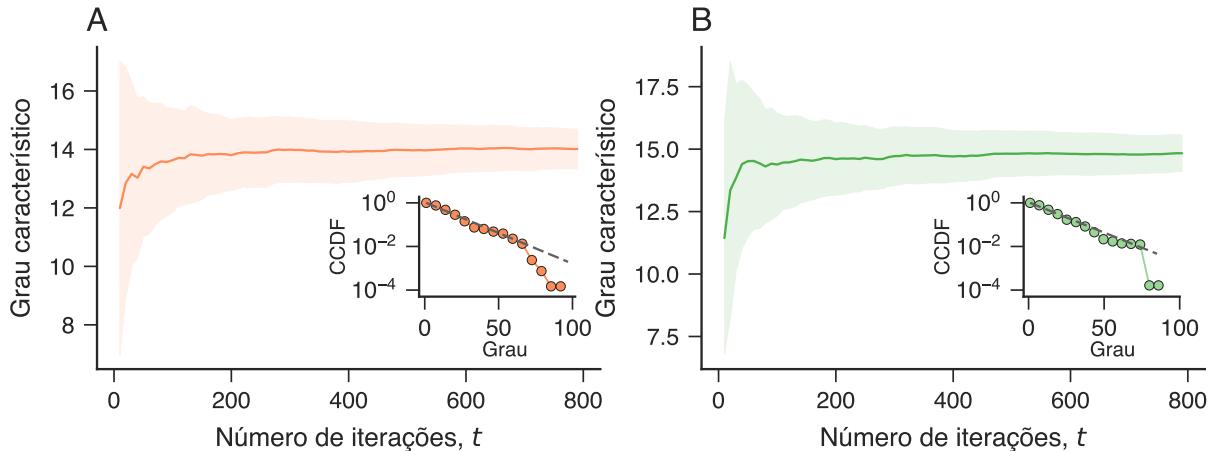


Figura 1.9: (A) Valor médio da estimativa de máxima verossimilhança do grau característico obtido por meio de 100 simulações usando a taxa de reincidência espanhola ($\alpha = 0.09$) em função do número de iterações t . A região sombreada representa intervalos de confiança de 95% obtidos via *bootstrap*. A inserção mostra a distribuição acumulada complementar de grau após $t = 800$ iterações do modelo. Nesse gráfico, a linha tracejada indica a distribuição exponencial ajustada aos dados. O painel (B) mostra a mesma análise com o mesmo número de simulações, mas usando a taxa de reincidência brasileira ($\alpha = 0.142$).

Quantitativamente, notamos uma diferença entre os valores de grau das distribuições empírica e teórica. A Figura 1.3 apresenta valores de grau maiores quando comparados aos valores presentes nas distribuições das redes simuladas da Figura 1.9. De modo geral, podemos dizer que as distribuições de grau das redes simuladas também apresentam boa concordância com as distribuições exponenciais, mas com graus característicos um pouco

menores que os empíricos (14.00 ± 0.66 versus 20.0 para a Espanha e 14.84 ± 0.72 versus 17.6 para o Brasil). Em parte, isso ocorre porque a distribuição exponencial do número de pessoas por caso, a qual consideramos para construir o modelo, subestima os dados para escândalos muito grandes (Figura 1.1).

Outro aspecto temporal importante que também encontramos nas redes simuladas é a existência de estruturas de comunidades, obtidas via o algoritmo *Infomap*. Além disso, de forma bastante similar ao comportamento empírico das Figuras 1.5A e 1.5B, observamos a presença de uma associação linear entre o número de módulos detectados e o número de grafos completos adicionados durante o processo de crescimento das redes. A Figura 1.10 ilustra esse aspecto linear com dois exemplos típicos de simulações de redes crescidas até 500 iterações e com taxas $\alpha = 0.090$ (Figura 1.10A) e $\alpha = 0.142$ (Figura 1.10B). Os coeficientes angulares dessas retas são, respectivamente, 0.813 ± 0.001 e 0.783 ± 0.001 ; valores esses ligeiramente maiores do que suas contrapartes empíricas (0.744 ± 0.004 para a Espanha e 0.630 ± 0.020 para o Brasil, respectivamente). Considerando esses valores obtidos, podemos concluir que as redes simuladas também exibem estruturas modulares que tendem a mesclar dois ou mais escândalos em módulos únicos.

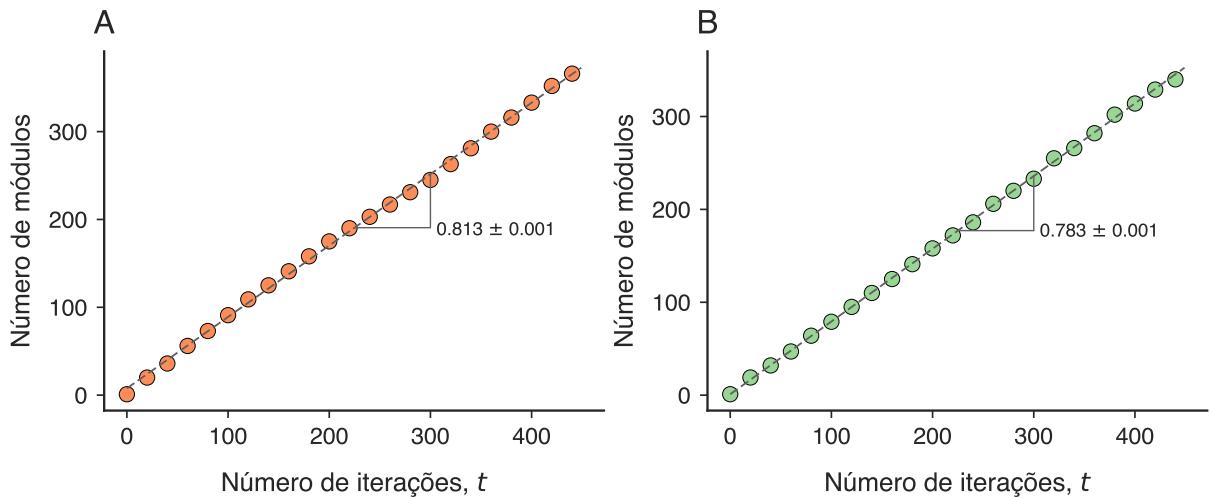


Figura 1.10: Os painéis (A) e (B) mostram exemplos típicos da associação linear entre o número de módulos e o total de escândalos para redes simuladas usando as taxas de reincidência espanhola e brasileira, respectivamente. Em ambos os painéis, as linhas tracejadas representam modelos lineares ajustados aos dados e os valores dentro desses painéis indicam o coeficiente angular dessas retas.

Ainda no contexto dinâmico das simulações, constatamos que as redes também apresentam coalescência de componentes durante seus crescimentos. Assim como no caso empírico (Figura 1.4), essas coalescências são verificadas por mudanças abruptas observadas no tamanho das maiores componentes. A Figura 1.11 exemplifica esse comportamento para redes geradas usando as taxas de reincidência da Espanha [$\alpha = 0.09$, painel (A)] e do Brasil [$\alpha = 0.142$, painel (B)]. Em ambos os painéis, observamos que o crescimento das

redes passa por um ponto no qual o tamanho da componente gigante aumenta substancialmente enquanto o tamanho da segunda maior componente diminui concomitantemente.

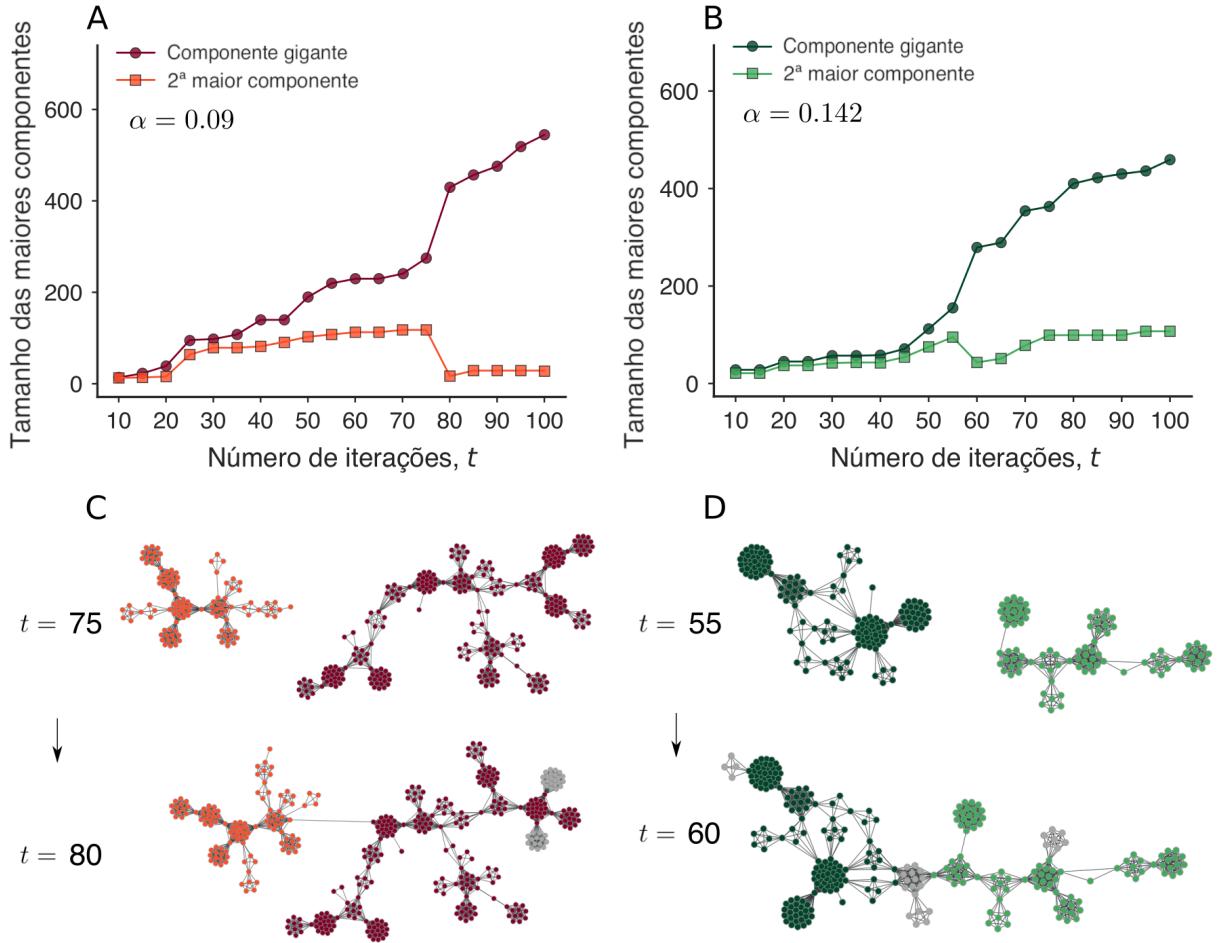


Figura 1.11: Os painéis (A) e (B) mostram exemplos típicos da evolução do tamanho das maiores componentes das redes simuladas usando as taxas de reincidência espanhola e brasileira, respectivamente. Observamos que essas redes também apresentam mudanças abruptas causadas pelo surgimento de novos escândalos envolvendo agentes reincidentes. Os painéis (C) e (D) mostram visualizações das redes simuladas antes e depois das mudanças abruptas observadas nos painéis (A) e (B), respectivamente. Nessas redes, novos vértices estão coloridos de cinza.

Destacamos essas ocorrências nas Figuras 1.11C e 1.11D, com visualizações do processo de coalescência entre as duas maiores componentes das redes simuladas. Na Figura 1.11C, de $t = 75$ para $t = 80$, alguns poucos agentes presentes na rede em $t = 75$ se tornam recorrentes no tempo $t = 80$. Isso faz com que ocorra a junção entre as duas maiores componentes da rede do tempo $t = 75$ no tempo $t = 80$. Similarmente, o mesmo processo pode ser observado na Figura 1.11D para simulações usando a taxa de reincidência da rede de corrupção brasileira. Apesar dessa figura apresentar apenas dois exemplos, esse comportamento esteve presente em diversas redes geradas usando nosso modelo. Considerando essa descoberta, podemos concluir que nosso modelo é capaz de reproduzir o processo de coalescência observado nas redes de corrupção empíricas. Nossos resultados

também indicam que os processos parecem sempre depender de um pequeno número de pessoas (ou vértices) que reincidem em diferentes escândalos (ou grafos completos). Esse fato reforça que reincidentes podem ser considerados de grande importância para evolução de redes de corrupção.

Em termos gerais, mostramos que é possível modelar o crescimento de redes de corrupção usando regras simples. O aspecto mais importante de nosso modelo é a taxa de reincidência, que não somente é responsável pela coesão entre as componentes da rede, mas também por seu comportamento evolutivo. Ao estudar as redes do modelo, encontramos a existência de uma taxa de reincidência crítica na qual o modelo passa a produzir redes com padrões de conexão mais complexos. Aplicando o modelo com as taxas de reincidência empíricas, conseguimos reproduzir todos os padrões empíricos.

Inicialmente, observamos que o modelo produz redes visualmente semelhantes às redes empíricas com medidas estruturais próximas aos valores empíricos. Em seguida, investigamos as distribuições de grau das redes geradas pelo modelo e concluímos que elas também podem ser razoavelmente bem descritas por distribuições de grau exponencial. Essas simulações apresentam padrões de evolução temporal do grau característico semelhante ao comportamento empírico. Além disso, as redes simuladas apresentam uma relação linear entre o número de módulos e o número de escândalos. Por fim, mostramos que a evolução das redes do modelo também é caracterizada por processos do tipo coalescência, nos quais grandes componentes das redes se conectam por meio de vértices reincidentes.

Embora a concordância entre as propriedades de redes empíricas e simuladas não seja perfeita, é surpreendente que um modelo tão simples reproduza qualitativamente várias características das redes de corrupção empíricas, incluindo propriedades dinâmicas. Parte das discrepâncias entre dados e modelo (como o menor grau característico e os maiores caminhos médios obtidos nas simulações) podem ser atribuídas aos desvios observados entre a distribuição exponencial e a distribuição de tamanho dos escândalos (Figura 1.1).

Uma possibilidade para futuras investigações é explorar o fato de que nosso modelo não faz distinção entre agentes corruptos além do aspecto da reincidência. Essa distinção é crucial no atual contexto de crescente polarização política, no qual se pode esperar que divisões partidárias e ideológicas também se refletem na corrupção política e, portanto, na estrutura das redes de corrupção. Além da provável importância desse e de outros mecanismos relacionados aos processos de corrupção, nossos resultados contribuem para o entendimento que a reincidência de uma pequena fração de agentes corruptos é crucial para a estrutura e dinâmica de redes de corrupção.

1.8 Robustez dos resultados

Uma questão que surge naturalmente em nosso trabalho se refere ao tamanho dos nossos conjuntos de dados. Assim como é provável que alguns escândalos de corrupção

e seus envolvidos não tenham sido descobertos (e, portanto, nossos conjuntos de dados estejam incompletos), podemos imaginar que o tamanho desses conjuntos poderiam ser ainda menores. Nesse contexto, é importante averiguar se nossas descobertas sobre as propriedades e dinâmicas das redes ainda seriam significativas caso nosso conjunto de dados fosse menor. Para testar a robustez dos nossos resultados, refizemos as análises anteriores com uma porcentagem dos dados (escândalos) removida. Para não adicionar nenhum viés nesse estudo, a escolha dos escândalos a serem removidos é feita de maneira aleatória e o processo é repetido diversas vezes para o cálculo da média e do desvio padrão referente à medida de interesse.

Nossa primeira investigação é em relação ao número característico de pessoas envolvidas em escândalos de corrupção. As Figuras 1.12A e 1.12B mostram, respectivamente, o valor médio do número característico de pessoas em escândalos de corrupção das redes espanhola e brasileira versus a fração de remoção de escândalos. Inicialmente, exibimos o resultado empírico das Figuras 1.1A e 1.1B (isto é, o valor característico da distribuição do tamanho dos escândalos) para obter uma referência. Depois, removemos 10% dos escândalos aleatoriamente, ajustamos uma nova distribuição exponencial aos novos dados e estimamos o novo valor característico de envolvidos por escândalo. Refazemos esse processo 100 vezes e obtemos o valor médio e o desvio padrão do conjunto (representado pelas barras de erro da Figura 1.12). O mesmo procedimento é realizado para a remoção de 20%, 30% e 40% dos casos de corrupção.

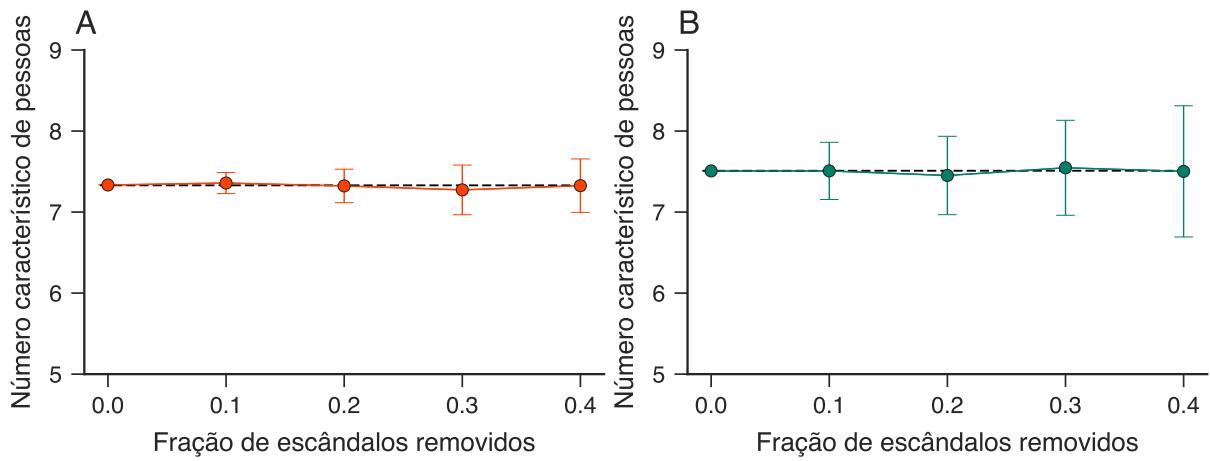


Figura 1.12: Estimativas do número característico de pessoas envolvidas em escândalos de corrupção na (A) Espanha e no (B) Brasil em função da fração de escândalos removidos. As linhas tracejadas indicam os valores ao considerar todos os escândalos em cada conjunto de dados. Os círculos representam valores médios estimados por meio de 100 realizações do processo de remoção aleatória e as barras de erro representam um desvio padrão desses valores. Em ambos os cenários, o número característico de pessoas permanece estável mesmo após a remoção de 40% dos escândalos.

Essa análise demonstra que o valor característico do número de pessoas por escândalo das redes de ambos os países varia pouco. Isto é, em todo processo (até 40% de casos

removidos), o número estimado fica próximo ao valor inicial e a barra de erro sempre compreende a linha tracejada que representa esse valor. Portanto, esse resultado sugere que os valores característicos do número de pessoas por escândalo (≈ 7.33 para a Espanha e ≈ 7.51 para o Brasil) são robustos contra a remoção aleatória de uma fração de casos dos nossos conjuntos de dados. Notamos que o desvio padrão é maior no cenário brasileiro (Figura 1.12B) para maiores frações. Isso se deve, parcialmente, ao menor tamanho do conjunto de dados brasileiro.

Nossa próxima investigação é em relação ao grau característico das redes de corrupção ao passo que removemos aleatoriamente uma fração dos escândalos. Queremos entender a robustez dos resultados presentes nas Figuras 1.3A e 1.3B. Para fazer isso, estimamos a distribuição de grau e o grau característico das redes ao passo que removemos os escândalos. As Figuras 1.13A e 1.13B mostram, respectivamente, o valor médio do grau característico das distribuições de grau das redes de corrupção espanhola e brasileira versus a fração de remoção de escândalos. Inicialmente, inserimos o resultado empírico das Figuras 1.3A e 1.3B (isto é, as estimativas do grau característico das distribuições de grau espanhola e brasileira) para obter uma referência. Depois, removemos 10% dos escândalos aleatoriamente, ajustamos uma nova distribuição exponencial aos novos dados e estimamos o novo grau característico. Refazemos esse processo 100 vezes e obtemos o valor médio e o desvio padrão desse conjunto (representado pelas barras de erro da Figura 1.13). O mesmo procedimento é realizado para a remoção de 20%, 30% e 40% dos casos de corrupção.

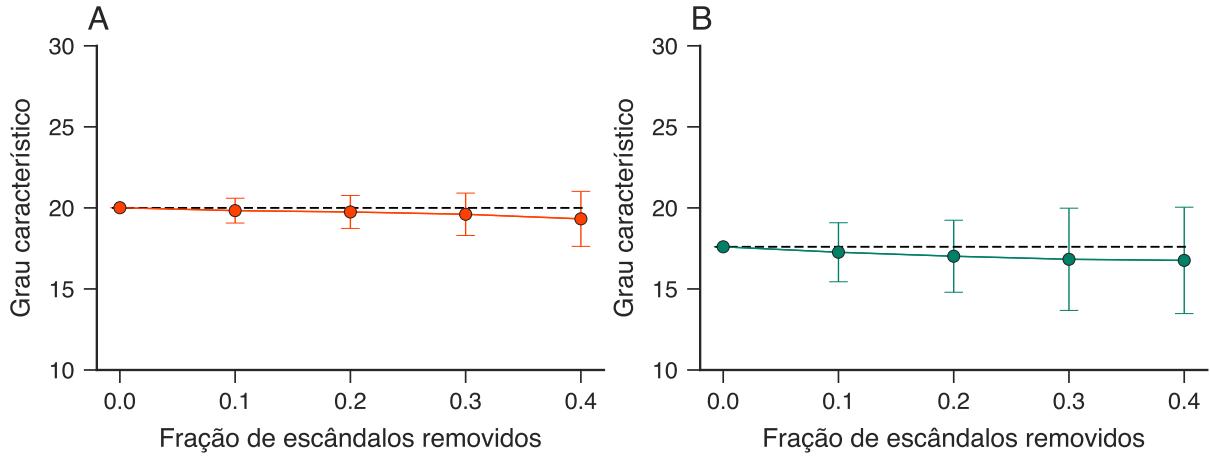


Figura 1.13: Estimativas do grau característico para a última etapa das redes de corrupção (A) espanhola e (B) brasileira em função da fração de escândalos removidos. As linhas tracejadas indicam os valores ao considerar as redes com todos os escândalos em nossos conjuntos de dados. Os círculos representam valores médios estimados via 100 realizações do processo de remoção aleatória e as barras de erro representam um desvio padrão desses valores. Em ambos os casos, o grau característico permanece estável mesmo após a remoção de 40% dos escândalos.

Observamos que o valor do grau característico das redes de ambos os países varia

pouco. Isto é, em todo processo (até 40% de casos removidos), o grau característico estimado fica bem próximo ao valor inicial e a barra de erro sempre compreende a linha tracejada que representa esse valor. Portanto, esse resultado sugere que o grau característico de redes de corrupção é robusto contra a remoção aleatória de uma fração de escândalos de nossos conjuntos de dados.

Em seguida, analisamos a robustez da associação linear entre o número de módulos da rede e seu número de escândalos. Para calcular quais são os novos módulos, novamente aplicamos o algoritmo *Infomap*. As Figuras 1.14A e 1.14B mostram as relações entre o número de módulos e o número de escândalos para cada ano das redes (A) espanhola e (B) brasileira sob diferentes frações de escândalos removidos. Inicialmente, exibimos o resultado empírico das Figuras 1.5A e 1.5B a fim de obter uma referência. Depois, removemos 10% dos escândalos aleatoriamente e crescemos a rede com o restante dos dados, calculando o número de módulos versus o número de escândalos a cada ano. Refazemos o mesmo processo removendo 20%, 30% e 40% dos casos de corrupção. Observamos que as curvas apresentam um aspecto linear para todas as frações. Notamos também que os desvios da linearidade são maiores para a rede brasileira. Parte dessa discrepância se deve ao fato de que o número de escândalos da rede espanhola (437) é significativamente maior do que o número de escândalos da rede brasileira (65) e, portanto, mesmo removendo até 40% dos casos espanhóis, esse conjunto de dados ainda permanece bastante representativo.

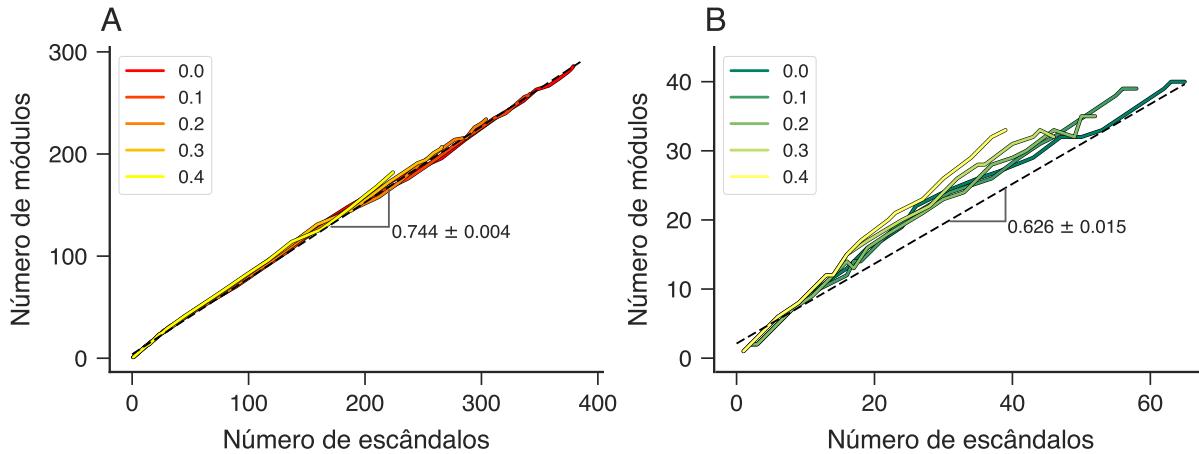


Figura 1.14: As curvas mostram as relações entre o número de módulos e o número de escândalos para cada ano das redes (A) espanhola e (B) brasileira sob diferentes frações de escândalos removidos (indicados pelo código de cores e pela legenda). As linhas tracejadas representam um modelo linear ajustado à relação obtida ao considerar todos os escândalos em cada conjuntos de dados.

Dando continuidade ao estudo da robustez de nossos resultados, examinamos a associação linear entre o número de reincidentes versus o número total de pessoas ao passo que removemos escândalos de corrupção. Novamente, tomamos como referência o resultado empírico com os dados completos (Figuras 1.5C e 1.5D). Depois, removemos 10% dos

escândalos aleatoriamente e crescemos a rede com o restante dos dados, calculando o número de pessoas reincidentes versus o número total de envolvidos a cada ano. Refazemos o mesmo processo removendo 20%, 30% e 40% dos casos de corrupção. As Figuras 1.15A e 1.15B mostram, respectivamente, essa relação para os dados das redes de corrupção espanhola e brasileira. Observamos que as curvas apresentam um aspecto linear para todas as frações e, novamente, os desvios do comportamento linear são mais pronunciados para a rede brasileira. Parte desse desvio pode ser atribuído à diferença entre o número de envolvidos reincidentes da Espanha (240) e do Brasil (56).

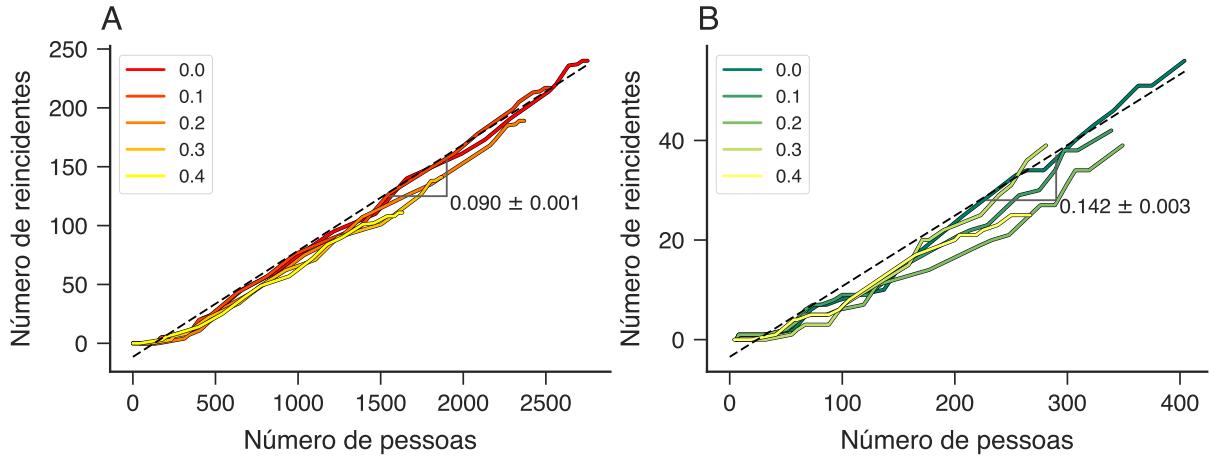


Figura 1.15: As curvas mostram a relação entre o número de agentes reincidentes e o número total de pessoas ao longo do crescimento das redes (C) espanhola e (D) brasileira sob diferentes frações de escândalos removidos (indicados pelo código de cores e pela legenda). As linhas tracejadas representam um modelo linear ajustado à relação obtida ao considerar todos os escândalos em cada conjuntos de dados.

De forma geral, os resultados apresentados anteriormente (Figuras 1.12, 1.13, 1.14 e 1.15) indicam que os padrões descobertos em nosso trabalho são robustos quanto a remoção aleatória de uma fração de escândalos.

1.9 Conclusões

Estudamos dois conjuntos de dados distintos e bem documentados sobre escândalos de corrupção política. Em um primeiro momento, exploramos o aspecto quantitativo desses escândalos e observamos que os envolvidos geralmente agem em grupos pequenos. Em ambos os conjuntos de dados, encontramos que o número de envolvidos por escândalo de corrupção é de aproximadamente 7 pessoas. Esse resultado sugere que escândalos em grande escala não são fáceis de administrar nem fáceis de executar.

Em seguida, analisamos os dados por uma perspectiva da ciência de redes. Construímos as redes de forma que os vértices representam indivíduos e ligações entre pares de vértices ocorrem caso duas pessoas estejam envolvidas ao menos em um mesmo escândalo.

Notamos que essas redes são compostas por uma componente gigante e vários outras componentes menores. Calculamos algumas medidas estruturais (coeficiente de agrupamento, coeficiente de assortatividade, densidade e comprimento de caminho médio) dessas redes e verificamos que elas possuem métricas bastante semelhantes. Ainda no contexto estrutural, encontramos estruturas de comunidades nas redes, com módulos que tendem a abranger mais de um escândalo de corrupção.

Ao considerar o aspecto temporal de nossos dados, também observamos similaridades no comportamento das redes. Mais especificamente, estudamos a evolução das distribuições de grau e encontramos que essas distribuições são bem descritas por modelos exponenciais. Durante o crescimento dessas redes, descobrimos a presença de tendências lineares em relação a determinadas variáveis. A primeira tendência linear identificada está relacionada ao número de comunidades em função do número de escândalos. Além disso, também observamos que o número de envolvidos que reincidem criminalmente cresce linearmente com o total de pessoas. Observamos ainda que a evolução temporal das redes de corrupção é caracterizada por mudanças abruptas no tamanho da maior componente, como consequência da coalescência de diferentes componentes conectadas.

Verificamos que os reincidentes apresentam um papel central na coesão e dinâmica das redes de corrupção. Considerando esses resultados, propomos um modelo para simular o crescimento de redes de corrupção, levando em conta principalmente o aspecto da reincidência. As redes geradas usando esse modelo produzem um comportamento emergente capaz de simular não somente redes estruturalmente semelhantes às suas contrapartes empíricas, mas também imitar seus padrões evolutivos.

Concentramos nossa atenção no comportamento das redes do modelo considerando a variação da taxa de reincidência. Concluímos que os processos de corrupção parecem operar próximos a uma taxa crítica de reincidência, abaixo da qual a rede se torna totalmente fragmentada e acima da qual ela é excessivamente conectada. Confirmamos a existência e o valor preciso dessa taxa crítica ao estudar o comportamento de evolução de redes ainda maiores.

Por fim, estudamos a robustez dos resultados em relação ao tamanho dos conjuntos de dados. Mostramos que o número característico de pessoas envolvidas nos escândalos das redes espanhola e brasileira varia pouco com a remoção de até 40% dos casos. Além disso, observamos que grau característico das redes também se mantém estável mesmo com a remoção de diversos casos. Também revelamos que a associação linear entre o número de módulos da rede e o número de escândalos, bem como a relação entre o número de reincidentes e o número total de envolvidos, são consistentes, sugerindo que os padrões descobertos são robustos contra a remoção aleatória de uma fração dos dados.

Capítulo 2

Aprendizagem de máquina aplicado a redes criminosas

Neste Capítulo, combinamos métodos de representação de grafos e aprendizagem de máquina em uma série de tarefas preditivas em redes criminosas [44]. Trabalhamos com quatro conjuntos de dados: duas redes de corrupção, uma rede de inteligência policial e uma rede relacionada a crimes financeiros.

Para abordar as tarefas, transformamos os elementos das redes em informações interpretáveis por algoritmos de aprendizagem de máquina. Utilizamos o método *Node2Vec* para gerar vetores associados aos vértices e, além disso, os combinamos por meio de operadores binários para obter vetores relacionados às ligações entre os vértices. De posse dessas informações, empregamos algoritmos de aprendizagem de máquina para tratar os problemas de classificação e regressão. Após separar os dados das redes em conjuntos de treinamento e teste, treinamos o modelo e avaliamos seu desempenho.

Na primeira seção, usamos classificadores logísticos para prever parcerias criminosas nas quatro redes criminosas. Investigamos o impacto de diferentes operadores e frações de ligações usadas no conjunto de treinamento, além de comparar os resultados com outros métodos de incorporação de vértices.

Em uma segunda aplicação, usamos classificadores k -primeiros vizinhos para distinguir entre diferentes tipos de ligações na rede de inteligência policial. Analisamos a influência do número de vizinhos k no desempenho dos classificadores, além de investigar como diferentes operadores binários e frações de ligações no conjunto treinamento afetam os modelos.

Na terceira seção, aplicamos regressores k -primeiros vizinhos para prever a quantidade de dinheiro trocada entre agentes da rede de crimes financeiros. Estudamos o desempenho do modelo comparando os valores previstos com os valores observados. Além disso, investigamos como o número de vizinhos k e diferentes frações de ligações no conjunto de treinamento afetam as previsões.

Por fim, abordamos o aspecto dinâmico das redes de corrupção e utilizamos classifica-

dores logísticos para prever futuras parcerias criminosas. Treinamos esses classificadores até um determinado ano das redes, gerando previsões de possíveis ligações e comparando-as com os dados dos anos subsequentes. Avaliamos o desempenho dos modelos ao longo do tempo, investigando a influência de diferentes algoritmos, métodos de incorporação e operadores binários.

2.1 Bases de dados

Nas próximas seções, utilizamos quatro redes criminosas provenientes de conjuntos de dados distintos. Duas dessas redes correspondem às redes previamente estudadas no Capítulo 1, relacionadas aos escândalos de corrupção na Espanha e no Brasil.

A terceira rede criminosa foi obtida a partir de uma parceria com a Polícia Federal Brasileira e compreende registros de investigações criminais conduzidas por essa força de segurança [23]. Os vértices dessa rede são criminosos ou suspeitos de atividades ilícitas relacionadas a crimes federais (por exemplo, tráfico de drogas e armas, assalto a banco organizado, crimes ambientais, crimes contra eleições e sistema financeiro, e lavagem de dinheiro) e conexões entre eles indicam indivíduos envolvidos no mesmo inquérito policial ou pessoas com relações pessoais descobertas durante as investigações. Essa rede de inteligência policial possui 23666 vértices e 35930 ligações. Além disso, para sua componente gigante (8894 vértices e 17827 ligações), temos informações sobre o tipo de associação entre os indivíduos, as quais estão classificadas em três tipos: criminosa, mista e não criminosa. As ligações criminosas conectam pessoas que se relacionam exclusivamente para fins ilícitos; ligações não criminosas conectam pessoas que não possuem associação criminosa e podem incluir laços familiares ou de amizade; finalmente, as conexões mistas representam associações criminosas e pessoais (por exemplo, dois irmãos envolvidos em uma investigação criminal).

O quarto conjunto de dados também foi obtido por meio da parceria com a Polícia Federal e está relacionado a uma investigação sobre lavagem de dinheiro realizada entre 2008 e 2014. Os dados brutos correspondem a transações bancárias relacionadas à apropriação indébita de recursos públicos federais. Após agregadas, essas informações geram uma rede financeira de 1126 vértices (representando pessoas ou empresas) e 1299 ligações (indicando transações financeiras).

2.2 Prevendo parcerias em redes criminosas

Iniciamos nossa investigação visando prever parcerias criminosas nas duas redes de corrupção e na rede de inteligência policial. A ideia é utilizar apenas informações estruturais dessas redes como dados de treinamento. Para isso, consideramos um cenário estático, no qual as redes de corrupção estão em seus estágios finais (isto é, com todos os escândalos

políticos) e a rede de inteligência policial também está completa. Os painéis (A), (B) e (C) da Figura 2.1 mostram visualizações de cada uma dessas redes.

O processo de previsão de ligações nas três redes é realizado da seguinte maneira. Inicialmente, removemos aleatoriamente 10% das ligações de cada rede e amostramos o mesmo número de ligações falsas para criar um conjunto de ligações verdadeiras e falsas. Esse é o conjunto de teste. Em seguida, usamos os 90% restantes das ligações de cada rede e amostramos o mesmo número de ligações falsas para criar o conjunto de treinamento. Dessa forma, mantemos os dados balanceados (isto é, cada conjunto possui a mesma proporção de ligações verdadeiras e falsas). Isso garante que a classificação não apresente viés para uma classe com maior proporção, uma vez que as classes possuem proporções iguais. Assim, podemos ajustar um modelo classificador para prever se as ligações no conjunto de teste são verdadeiras ou falsas.

Escolhemos o classificador logístico¹ como modelo para fazer essa tarefa de previsão. Para treinar esse classificador, precisamos obter representações vetoriais das ligações da rede. Assim, primeiramente, geramos uma representação vetorial para cada vértice usando o algoritmo *Node2Vec*². Em seguida, criamos as representações vetoriais das ligações (verdadeiras e falsas) combinando as representações vetoriais dos vértices por meio de quatro operadores binários: média, Hadamard e normas $L1$ e $L2$ ³. Dessa forma, construímos os conjuntos de treinamento e de teste, ambos contendo vetores que representam ligações verdadeiras e ligações falsas na mesma proporção.

Após treinar o classificador logístico por meio do conjunto de treinamento, podemos usá-lo para efetuar previsões usando o conjunto de teste. Mais especificamente, repetimos dez vezes o processo de gerar as representações vetoriais das ligações (verdadeiras e falsas) e dividir os dados em conjuntos de treinamento e de teste. Em cada etapa, calculamos e armazenamos a fração de classificações corretas (acurácia⁴) no conjunto de teste e depois obtemos o valor médio dessa medida.

A Figura 2.1D mostra as acurárias calculadas para as três redes usando os quatro operadores binários. Comparamos essas acurárias com um modelo de linha de base que escolhe aleatoriamente e com igual probabilidade se a ligação é verdadeira ou falsa. Uma vez que o conjunto de teste é balanceado (ou seja, mesma proporção de ligações verdadeiras e falsas), esse modelo produz uma acurácia de 50% (representada na Figura 2.1D pela linha horizontal tracejada de cor preta). Em todos os casos, a acurácia dos classificadores logísticos supera significativamente a acurácia do modelo de linha de base. Assim como nos

¹Descrevemos brevemente o algoritmo do classificador logístico no Apêndice C.3.

²O Apêndice C.10 apresenta o processo de transformação de vértices em vetores pelo algoritmo *Node2Vec*. Nesse Capítulo, fixamos a dimensão dos vetores em $d = 256$, o comprimento de cada caminhada (em número de vértices) igual a 5 e o número de caminhadas por vértice igual a 10. Além disso, usamos os parâmetros de viés da caminhada aleatória (p e q) iguais a 1, tornando o algoritmo de incorporação semelhante ao método original *DeepWalk*.

³Definimos cada um desses operadores no final do Apêndice C.10.

⁴O Apêndice C.5 apresenta detalhes sobre a medida de acurácia.

estudos da referência [37], nossos resultados também mostram que o operador Hadamard apresenta o melhor desempenho nas três redes criminosas. A acurácia do modelo com esse operador é de $\approx 98\%$ na rede de corrupção espanhola, $\approx 96\%$ na rede de corrupção brasileira e $\approx 87\%$ na rede de inteligência policial.

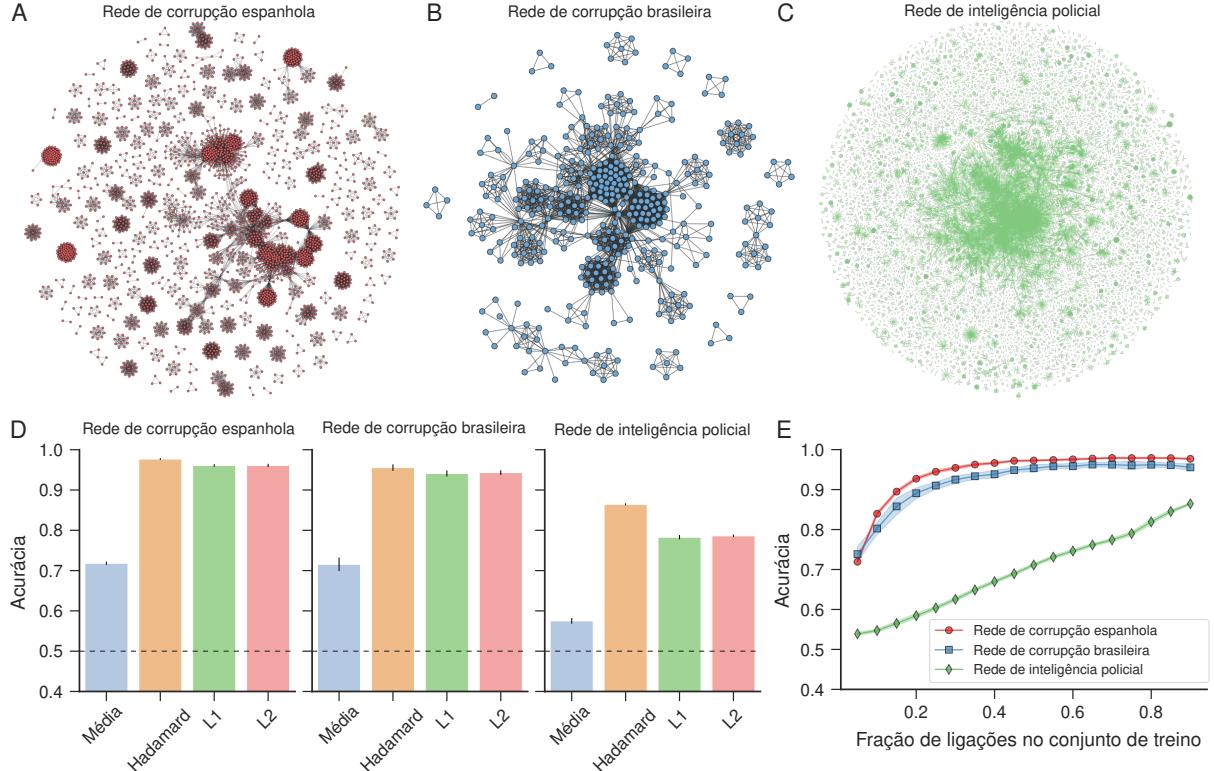


Figura 2.1: A primeira linha de painéis mostra uma visualização comparativa entre as redes de corrupção (A) espanhola e (B) brasileira (já introduzidas no Capítulo 1), e a rede de inteligência policial (C). (D) Acurácia do classificador logístico usando diferentes operadores binários. As barras representam a acurácia média estimada nos conjuntos de teste via dez realizações do processo de incorporação e treinamento. Nesse painel, as barras de erro representam um desvio padrão. As linhas tracejadas horizontais representam a acurácia (0.5) de um modelo de linha de base no qual as previsões são realizadas aleatoriamente. (E) Acurácia do classificador logístico para cada rede criminosa em função da fração dos vértices usados no conjunto de treinamento. Os marcadores representam a acurácia média estimada nos conjuntos de teste via dez realizações do processo de incorporação e treinamento com o operador Hadamard. Nesse painel, as bandas coloridas representam um desvio padrão.

Além da análise anterior, investigamos como o desempenho de nossa abordagem depende da fração de ligações usadas para treinar o classificador logístico. Para tanto, realizamos o mesmo procedimento descrito anteriormente variando a porcentagem de ligações presentes no conjunto de treinamento (e, consequentemente, no conjunto de teste). A Figura 2.1E mostra a acurácia do algoritmo em função da fração de ligações usadas para treinar o modelo em cada uma das três redes. Observamos que as acurárias nas redes de corrupção se aproximam de seus valores máximos muito mais rapidamente do que a

acurácia na rede de inteligência policial. Por exemplo, não observamos praticamente nenhuma mudança nas acurárias das redes de corrupção após considerar 60% das ligações no conjunto de treinamento. Por outro lado, essa medida para a rede de inteligência policial aumenta monotonicamente com a fração de ligações usadas no processo de incorporação.

Esses resultados indicam que a estrutura das redes de corrupção é mais redundante do que a estrutura da rede de inteligência policial. De fato, mostramos no Capítulo 1 que as redes de corrupção são formadas por grafos completos representando escândalos de corrupção, no qual todos os envolvidos estão conectados. Esses escândalos, por sua vez, estão interligados devido à reincidência criminosa de um pequeno número de agentes. Por conseguinte, acreditamos que o classificador logístico é capaz de capturar esse padrão por meio das representações vetoriais dos vértices. Em contraste, a rede de inteligência policial possui padrões de conexões mais complexos visto que as investigações policiais não produzem, obrigatoriamente, grafos completos na rede. Consequentemente, é mais difícil para o método de incorporação codificar essas relações e o classificador realizar boas previsões.

O *Node2Vec* não é o único algoritmo capaz de produzir representações vetoriais para os vértices de uma rede. De fato, como iremos também explorar no próximo Capítulo, existem muitas abordagens para realizar essa tarefa. Aqui, comparamos o desempenho do *Node2Vec* com os outros dois algoritmos de incorporação comumente usados em combinação com classificadores logísticos: *LINE* e *Mercator*⁵.

A Figura 2.2 compara a acurácia do classificador logístico aplicado aos dados das redes criminosas usando os três métodos de incorporação: *Node2Vec* (Figura 2.2A), *LINE* (Figura 2.2B) e *Mercator* (Figura 2.2C). Para cada rede, geramos as representações de todos os vértices por meio de cada um desses métodos. Em seguida, criamos os conjuntos de treinamento e de teste seguindo o mesmo procedimento descrito anteriormente, no qual removemos aleatoriamente 10% das ligações da rede e amostramos o mesmo número de conexões falsas. Após isso, obtemos as representações vetoriais de todas as ligações por meio dos operadores média, Hadamard, L1 e L2.

De posse dessas informações, podemos treinar o classificador logístico usando o conjunto de treinamento e estimar sua acurácia no conjunto de teste. Similarmente à análise da Figura 2.1D, repetimos dez vezes o processo de gerar os vetores das ligações (verdadeiras e falsas) e dividir os dados nos conjuntos de treinamento e de teste. Em cada etapa, armazenamos a fração de classificação correta (acurácia) do classificador no conjunto de teste. As barras da Figura 2.2 representam os valores médios das acurárias e as barras de erro representam um desvio padrão. Notamos que as acurárias dos classificadores com as abordagens de incorporação *LINE* e *Mercator* também superam a acurácia do modelo de linha de base. No entanto, esses valores são sempre menores do que as médias obtidas ao

⁵Os Apêndices C.11 e C.12 apresentam, respectivamente, os algoritmos *LINE* e *Mercator* com maiores detalhes.

usar o *Node2Vec* para gerar os vetores iniciais.

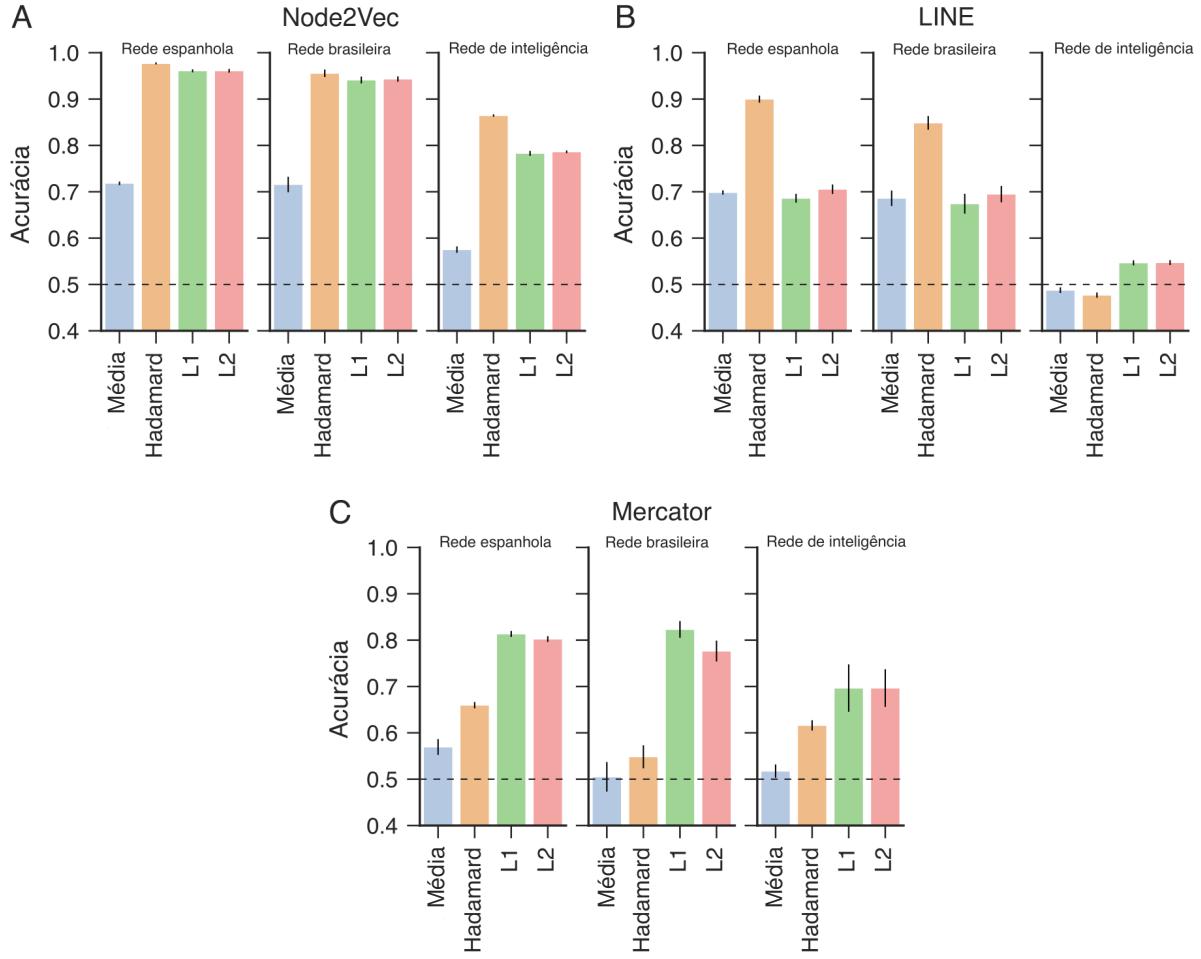


Figura 2.2: Acurácia média de classificadores logísticos treinados para prever ligações nas redes criminosas usando vértices de entrada obtidos via (A) *Node2Vec*, (B) *LINE* e (C) Mercator. Os gráficos de barra representam a acurácia média nos conjuntos de teste via dez repetições dos processos de incorporação e treinamento. Nesses painéis, as barras de erro representam um desvio padrão e as linhas tracejadas horizontais indicam a acurácia de um modelo de linha de base que prevê aleatoriamente se a ligação é verdadeira ou falsa.

2.3 Classificando a associação entre envolvidos da rede de inteligência policial

A combinação de vetores associados aos vértices de uma rede nos possibilita prever outras variáveis além da existência de ligações. Por exemplo, caso exista alguma característica associada à cada ligação, podemos treinar um modelo para aprender a relação entre os vetores das ligações e suas características. Nesse contexto, uma tarefa interessante é tentar usar a estrutura da rede de inteligência policial para determinar o tipo de associação (criminosa, mista ou não criminosa) entre dois envolvidos. O foco da nossa

análise é a maior componente (com 8894 vértices e 17827 ligações) dessa rede, sobre a qual possuímos essas informações extras. A Figura 2.3A mostra uma visualização dessa componente, na qual os três tipos de associações (criminosas, mistas e não criminosas) estão representadas em cores diferentes (vermelho, azul e verde, respectivamente).

Começamos nossa análise gerando um vetor para cada vértice por meio do *Node2Vec*. Para obter os vetores das ligações, combinamos os vetores dos vértices mediante os operadores binários média, Hadamard, L1 e L2 usados anteriormente. Em seguida, separamos (mantendo as mesmas proporções de exemplos das três classes)⁶ 10% das ligações para o conjunto de teste e utilizamos os 90% restantes como conjunto de treinamento. Para o conjunto de treinamento, visto que as classes das ligações são desproporcionais (54% criminosas, 22% mistas e 24% não criminosas), usamos a estratégia de amostragem aleatória a fim de replicar aleatoriamente os exemplos de classes minoritárias [56]. Dessa forma, aumentamos o número de exemplos das classes mista e não criminosa para que as mesmas apresentem o mesmo número de exemplos que a classe criminosa no conjunto de treinamento.

Para realizar a tarefa de classificação, ajustamos o classificador *k*-primeiros vizinhos⁷ (*kNN*) aos dados de treinamento e estimamos sua acurácia média no conjunto de teste por meio de dez realizações dos procedimentos de obtenção dos vetores e treinamento. Nesse processo, o número de vizinhos foi tomado como $k = 1$. A Figura 2.3B mostra os valores da acurácia média para cada operador binário. Para fins de comparação, mostramos os resultados de dois classificadores simples: o primeiro realiza previsões com base na frequência relativa de cada tipo de ligação (linha cinza contínua) e o segundo sempre escolhe o tipo de ligação mais frequente (linha preta tracejada). Observamos que a acurácia do classificador via qualquer um dos operadores binários é significativamente maior do que a acurácia desses dois classificadores simples. É importante também destacar que o operador Hadamard novamente exibe a maior acurácia (74%), seguido do operador média.

A Figura 2.3C exibe a matriz de confusão⁸ da tarefa de classificação dos tipos de associações. Os valores dessa matriz representam a fração média de acerto para cada associação. Essa média é estimada no conjunto de teste por meio de dez realizações dos processos de obtenção dos vetores (usando o operador Hadamard) e treinamento. Observamos que identificar ligações mistas é mais desafiador para o algoritmo *kNN*, uma vez que ele classifica corretamente esse tipo de ligações em 55% dos casos. Em contraste, ligações criminosas e não criminosas são classificadas corretamente 81% e 77% das vezes, respectivamente. Também é importante notar que o algoritmo classifica erroneamente as relações mistas como ligações criminosas com mais frequência do que relações não criminosas, o que pode ser considerado uma propriedade interessante se lembrarmos que esse tipo de

⁶Em aprendizagem estatística, esse procedimento é conhecido como estratificação dos dados.

⁷O Apêndice C.4 apresenta brevemente o método dos *k*-primeiros vizinhos.

⁸O Apêndice C.6 apresenta os principais aspectos de uma matriz de confusão e também detalha sua interpretação.

associação está sempre relacionada a um possível crime. No próximo Capítulo, veremos que essas classificações podem ser melhoradas significativamente com o uso de métodos de aprendizagem de máquina mais sofisticados.

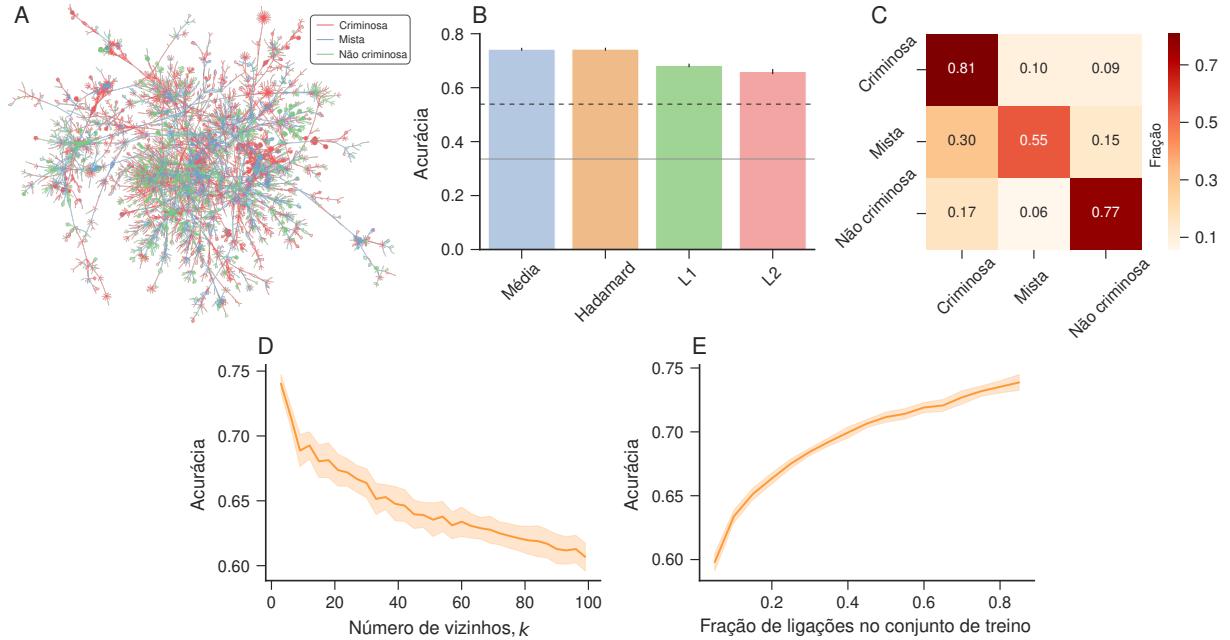


Figura 2.3: (A) Visualização da maior componente da rede de inteligência policial com destaque para os três tipos de associação entre os agentes. As ligações em vermelho, azul e verde representam relações criminosas, relações mistas e relações não criminosas, respectivamente. (B) Acurácia média dos classificadores kNN (com $k = 1$) treinados com vértices obtidos via *Node2Vec* e diferentes operadores binários. As barras representam a acurácia média estimada no conjunto de teste por meio de dez realizações dos processos de obtenção dos vetores e treinamento. As barras de erro representam um desvio padrão. A linha cinza contínua representa a acurácia de um classificador simples que faz previsões aleatórias com base na frequência relativa de cada tipo de associação no conjunto de treinamento. Por outro lado, a linha preta tracejada indica a acurácia de um classificador que sempre prevê o tipo mais comum de associação (isto é, ligação criminosa). (C) Matriz de confusão relacionada às previsões do classificador kNN com $k = 1$ e usando o operador Hadamard (as linhas horizontais da matriz indicam a associação verdadeira e as linhas verticais indicam a associação prevista pelo classificador). (D) Acurácia média do classificador kNN em função do número de vizinhos (k). (E) Acurácia média do classificador kNN em função da fração de ligações nos conjuntos de treinamento. Nesses dois últimos painéis, as linhas laranjas indicam a acurácia média e as regiões sombreadas representam uma banda de desvio padrão estimada por meio de dez realizações dos processos de obtenção dos vetores e treinamento com o operador Hadamard.

Outro aspecto do nosso estudo que merece destaque é entender como o número de vizinhos (k) do classificador kNN pode afetar a acurácia na determinação do tipo de associação. A Figura 2.3D mostra a acurácia média em função do número de vizinhos. Estimamos a acurácia no conjunto de teste por meio de dez realizações dos processos de obtenção dos vetores (usando o operador Hadamard) e treinamento. Observamos que

valores mais altos são obtidos para um número pequeno de vizinhos e que a acurácia diminui com o aumento de k . Os resultados apresentados nas Figuras 2.3B e 2.3C são obtidos usando $k = 1$, uma vez que esse valor produz a maior acurácia.

Por fim, também verificamos como a acurácia depende da fração de ligações utilizadas para treinar o classificador kNN . Para fazer isso, consideramos uma fração variável de ligações (X) para treinar o kNN e usamos o restante ($1 - X$) das ligações como conjunto de teste. A Figura 2.3E mostra a curva da acurácia média em função da fração do conjunto de treinamento. Aqui, novamente estimamos a acurácia no conjunto de teste por meio de dez realizações dos processos de obtenção dos vetores (usando o operador Hadamard) e treinamento. Notamos que a acurácia média aumenta monotonicamente com a fração de ligações no conjunto de treinamento. No entanto, a mudança na acurácia é maior quando consideramos menores frações de ligações usadas durante treinamento.

2.4 Prevendo valores de transações bancárias na rede de crime financeiros

Em nosso próximo estudo, desejamos novamente prever um atributo associado a cada ligação. O foco de nossa análise é a rede de lavagem de dinheiro, na qual as ligações correspondem a transações bancárias entre os envolvidos (vértices). Nessa rede, buscamos estimar o logaritmo da quantidade de dinheiro trocado entre seus agentes com base exclusivamente nas informações estruturais obtidas via *Node2Vec*. Considerando que os atributos que queremos prever são variáveis contínuas, estamos tratando de uma tarefa de regressão.

A Figura 2.4A mostra uma visualização dessa rede, a qual possui 1126 vértices e 1299 ligações. A espessura das ligações é proporcional ao logaritmo da quantidade de dinheiro trocado entre os pares de vértices. Além disso, quanto mais clara a cor da ligação, maior a quantidade de dinheiro trocada entre os agentes.

Por meio de uma abordagem similar àquela que adotamos na rede de inteligência policial, usamos o *Node2Vec* para gerar os vetores de todos os vértices da rede de crimes financeiros. Assim, aplicamos os quatro operadores binários para combinar esses vetores e produzir os vetores associados às ligações. Em seguida, construímos um conjunto de dados associando o vetor de cada ligação ao logaritmo da quantidade de dinheiro envolvida. Feito isso, dividimos os dados em conjuntos de treinamento (90%) e teste (10%).

Para realizar a tarefa de regressão, escolhemos o algoritmo de regressão kNN devido à sua simplicidade. Treinamos esse regressor para prever o logaritmo da quantidade de dinheiro e estimamos o desempenho de nossa abordagem calculando o coeficiente de determinação⁹ (R^2) entre os valores previstos e observados (isto é, presentes no conjunto de

⁹O Apêndice C.7 detalha o cálculo desse coeficiente.

teste). Além disso, calculamos a média desse coeficiente via dez realizações dos processos de obtenção dos vetores, treinamento e teste. Aqui, usamos o algoritmo *Node2Vec* para gerar os vetores e, para combiná-los, aplicamos os quatro operadores usados anteriormente. A Figura 2.4B mostra o valor médio de R^2 obtido para cada operador binário. Para fins de comparação, empregamos dois regressores simples, sendo que o primeiro sempre retorna a média (linha preta tracejada) dos valores do conjunto de treinamento e o segundo sempre retorna a mediana (linha cinza contínua) desse conjunto.

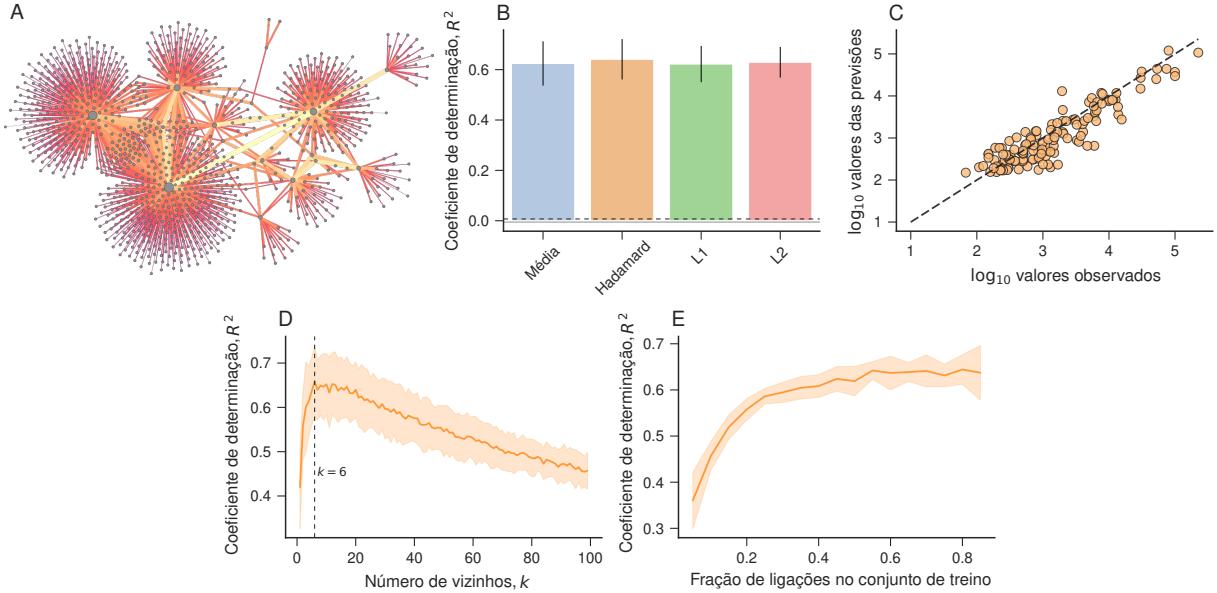


Figura 2.4: (A) Visualização da rede de crimes financeiros. Os vértices representam agentes (pessoas ou empresas) e as ligações indicam transações financeiras. Quanto mais grossa a ligação e mais clara sua cor, maior o valor da transação. (B) Coeficiente de determinação (R^2) médio da associação entre as transações (em valores logarítmicos) previstas e observadas. Essas previsões são obtidas usando regressores kNN com o número de vizinhos $k = 6$ treinados com vetores gerados via *Node2Vec* e combinados com diferentes operadores binários. As barras representam a acurácia média e as barras de erro representam um desvio padrão. Essas estatísticas são obtidas via dez realizações dos processos de obtenção dos vetores e treinamento. Para fins de comparação, a linha cinza contínua representa a acurácia de um regressor que sempre retorna o valor médio do conjunto de treinamento e a linha tracejada preta representa a acurácia de outro regressor que sempre retorna a mediana do conjunto de treinamento. (C) Exemplo típico da relação entre o logaritmo das quantidades de dinheiro previstas e observadas nas ligações dos conjuntos de teste. Esse exemplo é obtido usando um regressor kNN ($k = 6$) treinado com vetores gerados via *Node2Vec* e combinados por meio do operador Hadamard. A linha tracejada representa a relação 1:1. (D) Valor médio do coeficiente R^2 em função do número de vizinhos (k) do regressor kNN . A linha tracejada vertical indica o número ideal de vizinhos ($k = 6$). (E) Valor médio do coeficiente R^2 em função da fração de ligações no conjunto de treinamento. Nos dois últimos painéis, as linhas laranjas indicam o valor médio de R^2 e as regiões sombreadas representam uma banda de desvio padrão. Essas estatísticas são estimadas por meio de dez realizações do processo de obtenção dos vetores (com *Node2Vec* e operador Hadamard) e treinamento.

Observamos que o desempenho do regressor kNN é muito melhor do que desses modelos simples, produzindo valores de R^2 em torno de 0.6 para todos os operadores binários. A Figura 2.4B também revela que, novamente, o operador Hadamard exibe o desempenho médio mais alto ($\approx 0.64\%$) dentre os quatro operadores binários, embora a diferença entre eles seja pequena. Para ilustrar o procedimento anterior de obtenção do coeficiente de determinação, separamos um exemplo da previsão do modelo com o operador Hadamard. A Figura 2.4C mostra uma relação típica entre os valores (em logaritmo de base 10) previstos e observados. A linha tracejada exibe a relação de 1:1 e representa o caso perfeito no qual os valores previstos pelo regressor são exatamente aqueles observados nos dados. Esse exemplo destaca que nosso modelo possui uma boa capacidade de previsão.

De forma similar ao estudo que fizemos sobre o classificador para prever associações criminosas, investigamos como o número de vizinhos (k) afeta a acurácia do modelo. Para fazer isso, novamente separamos os dados em conjuntos de treinamento (90%) e teste (10%), aplicamos o *Node2Vec* seguido do operador Hadamard e depois estimamos o coeficiente R^2 do regressor para cada valor de k . A Figura 2.4D mostra essa análise. A linha laranja indica o valor médio do coeficiente e a região sombreada representa uma banda de desvio padrão estimada via dez realizações do processo de obtenção de R^2 . Observamos que $k = 6$ (indicado na figura pela linha vertical tracejada) produz o maior desempenho do modelo. Por esse motivo, usamos $k = 6$ para obter os resultados anteriores (Figuras 2.4B e 2.4C).

Para completar nossa análise, investigamos como a fração de ligações no conjunto de treinamento afeta o coeficiente R^2 obtido no conjunto de teste. Para obter os valores de R^2 , realizamos o mesmo processo descrito anteriormente para dividir, treinar e testar o modelo (aqui também usamos o operador Hadamard). A Figura 2.4E mostra o valor médio de R^2 em função da fração de ligações no conjunto de treinamento. Notamos que esse coeficiente satura aproximadamente após considerar um pouco mais de 50% das ligações.

2.5 Prevendo parcerias futuras nas redes de corrupção

Como última aplicação desse Capítulo, consideramos o problema mais desafiador de prever parcerias futuras usando a estrutura das redes criminosas. Focamos nossa análise nas duas redes de corrupção porque são as únicas que possuímos a dinâmica de crescimento. Como já investigamos, essas redes crescem no tempo devido a novos escândalos envolvendo réus primários e reincidentes, sendo esses últimos responsáveis por interligar os diferentes casos de corrupção.

Para abordar esse problema, consideramos escândalos ocorridos até um determinado ano Y para construir a rede criminosa G_Y . Aplicamos o *Node2Vec* nessa rede para gerar

os vetores associados a todos os vértices. Para cada um dos quatro operadores binários, produzimos os vetores associados a todas as ligações da rede. Além disso, amostramos aleatoriamente o mesmo número de ligações falsas para completar o conjunto de treinamento. Com esse conjunto, treinamos um classificador logístico para distinguir entre ligações verdadeiras e falsas.

O conjunto de teste, por sua vez, é construído da seguinte maneira. Conferimos todos os escândalos de corrupção ocorridos após o ano Y e armazenamos apenas as ligações entre vértices que já estavam presentes em G_Y . Essas ligações representam futuras parcerias criminosas entre vértices de G_Y . Além disso, amostramos aleatoriamente a mesma quantidade de ligações falsas que não ocorrem no futuro de G_Y . Consideramos os vetores dos vértices que obtemos em G_Y para gerar (via cada operador binário) os vértices das ligações futuras verdadeiras e falsas. Observe que nenhuma informação sobre escândalos ocorridos após o ano Y é usada para criar os vetores das ligações no conjunto de teste ou para treinar o modelo logístico.

Aplicamos o classificador logístico para determinar se as conexões no conjunto de teste são verdadeiras ou falsas e estimamos a acurácia média de nossa abordagem via dez realizações de todo o processo de treinamento, incorporação e classificação. O painel central da Figura 2.5 mostra a acurácia média¹⁰ nos conjuntos de teste ao considerar diferentes anos limites (Y) para as redes de corrupção espanhola (círculos vermelhos) e brasileira (quadradinhos azuis). As inserções indicadas por setas exibem visualizações de G_Y para alguns anos, nas quais ligações em cinza indicam futuras parcerias criminosas. Essas inserções também mostram exemplos da matriz de confusão do processo de classificação.

Observamos que os classificadores logísticos apresentam acuráncias superiores a 0.8 para a maioria dos anos na rede de corrupção espanhola. Esses valores superam significativamente a acurácia de 0.5 de um classificador simples (de linha de base) que escolhe aleatoriamente (e com igual probabilidade) se a ligação é verdadeira ou falsa. Por outro lado, na rede de corrupção brasileira, observamos que para anos anteriores a 2003, a acurácia média do classificador não difere muito da acurácia do modelo de linha de base. No entanto, após esse ano as médias flutuam em torno de 0.65 e superam significativamente a acurácia do modelo de linha de base.

Para fins de comparação, queremos entender o desempenho (também em termos da acurácia média) da tarefa de classificação de futuras ligações ao usar diferentes operadores binários e classificadores. Primeiramente, consideramos a rede de corrupção brasileira. A Figura 2.6A mostra os valores da acurácia do classificador logístico para diferentes operadores binários. Em seguida, mostramos a acurácia de classificadores k -primeiros vizinhos com $k = 3, 5, 7, 15$ e 50 (Figuras 2.6B, 2.6C, 2.6D, 2.6E e 2.6F, respectivamente). Essas

¹⁰Para obter os resultados presentes na Figura 2.5, usamos o operador Hadamard para a rede espanhola e o operador média para a rede brasileira. Como veremos mais adiante, essas escolhas resultam nos maiores valores de acurácia média.

figuras também mostram a acurácia ao usar cada um dos operadores binários. No geral, notamos que a melhor combinação de classificador e operador (ou seja, aquela que resulta no melhor desempenho) é o classificador logístico com o operador média. É por essa razão específica que o resultado presente na Figura 2.5 usa essa combinação de classificador e operador para a rede de corrupção brasileira.

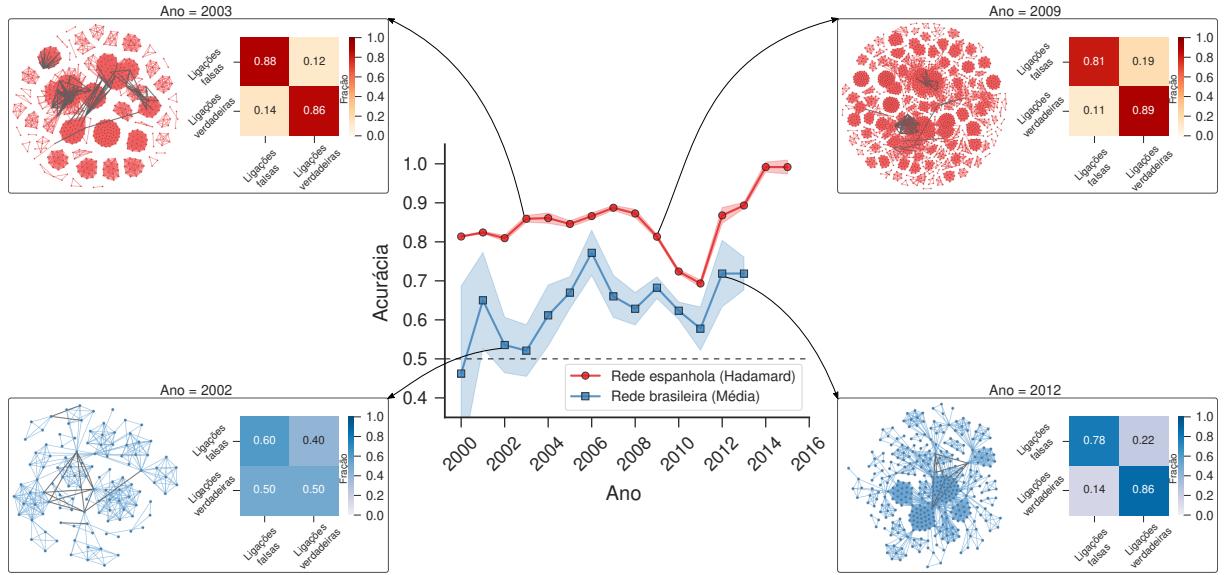


Figura 2.5: O painel central mostra a acurácia média da tarefa de prever ligações futuras nas redes de corrupção espanhola (círculos vermelhos) e brasileira (quadrados azuis). As acurárias são calculadas considerando a rede com escândalos ocorridos até determinado ano. Os marcadores representam a acurácia média para diferentes anos limite. Esses valores médios são obtidos nos conjuntos de teste estimados por meio de dez realizações do processo de obtenção dos vetores e treinamento. Nesse painel, as regiões sombreadas representam uma banda de um desvio padrão. A linha tracejada preta indica a acurácia do modelo de linha de base. As inserções mostram visualizações de redes nas quais as ligações coloridas representam conexões entre os vértices que ocorreram até o ano limite, enquanto as ligações cinza representam ligações que aparecerão após esse ano. Essas inserções também acompanham matrizes de confusão associadas às tarefas de prever se as ligações futuras são verdadeiras ou falsas. Nessa matrizes, as linhas horizontais indicam o observado e as linhas verticais indicam a previsão do classificador.

Realizamos a mesma comparação de classificadores e operadores binários para a rede de corrupção espanhola. A Figura 2.7A mostra os valores da acurácia do classificador logístico para diferentes operadores binários. Para cada operador, também mostramos o desempenho de classificadores k -primeiros vizinhos com $k = 3, 5, 7, 15$ e 50 (Figuras 2.7B, 2.7C, 2.7D, 2.7E e 2.7F, respectivamente). Nessas figuras, observamos que o classificador logístico com o operador Hadamard apresenta, em geral, uma melhor acurácia. Portanto, usamos essa combinação de classificador e operador para produzir o resultado da tarefa de previsão na rede de corrupção espanhola (Figura 2.5).

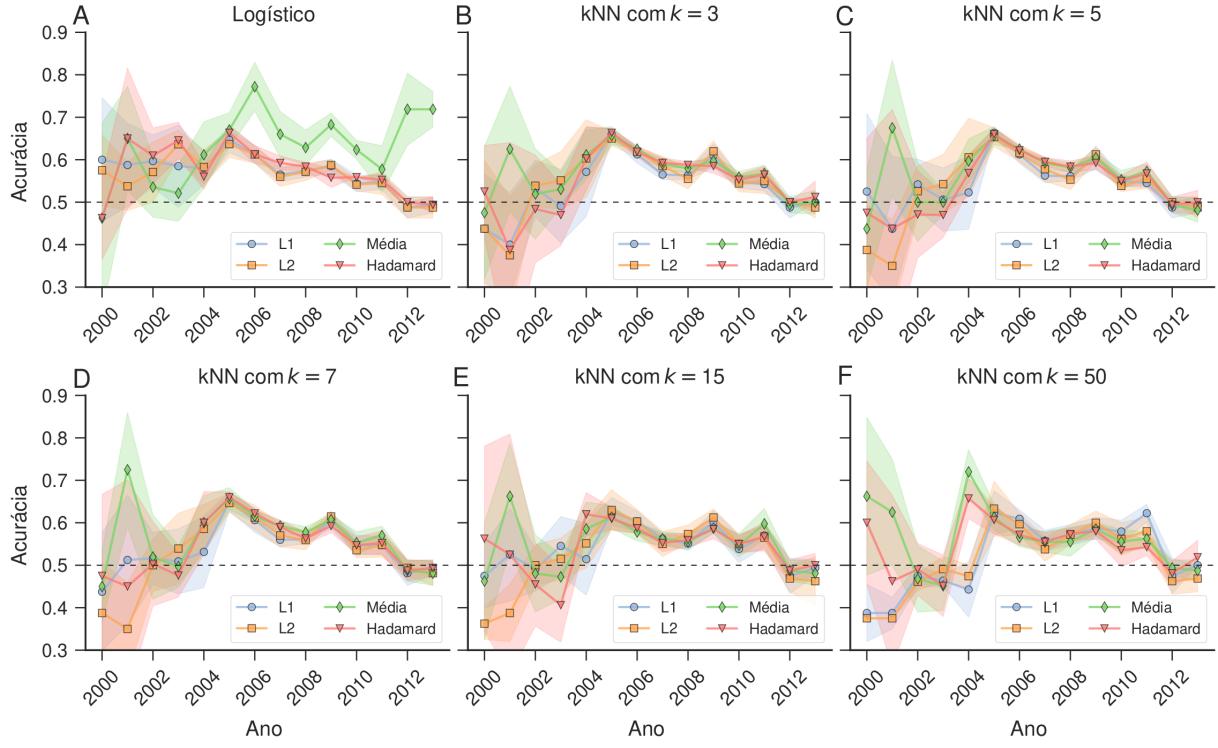


Figura 2.6: Acurácia média da previsão de parcerias futuras na rede de corrupção brasileira em função do ano limiar para diferentes operadores binários. O painel (A) mostra os resultados obtidos usando classificadores logísticos, enquanto os outros painéis [(B), (C), (D) e (E)] mostram a acurácia para classificadores kNN com diferentes números de vizinhos. Os marcadores representam a acurácia média nos conjuntos de teste estimados a partir de dez realizações do processo de obtenção das representações dos vértices e treinamento (as regiões sombreadas representam uma banda de desvio padrão). A linha tracejada horizontal indica a acurácia do modelo de linha de base.

Os gráficos das Figuras 2.6 e 2.7 revelam que o classificador logístico possui o melhor desempenho em ambas as redes de corrupção. Além disso, a tarefa de previsão na rede de corrupção espanhola possui maior acurácia do que na rede de corrupção brasileira. Como mostramos no Capítulo 1, tanto a estrutura quanto a evolução dessas redes possuem similaridades. De fato, revelamos ser possível até mesmo replicar seus padrões por meio de um modelo computacional simples. Dessa forma, é razoável imaginar que a acurácia desses classificadores deveriam ser similares. No entanto, a acurácia é menor na rede de corrupção brasileira. Nesse contexto, acreditamos que essa diferença é parcialmente causada pelo fato de que a rede de corrupção espanhola é maior e, portanto, o classificador possui mais informações para aprender seus padrões de conexões.

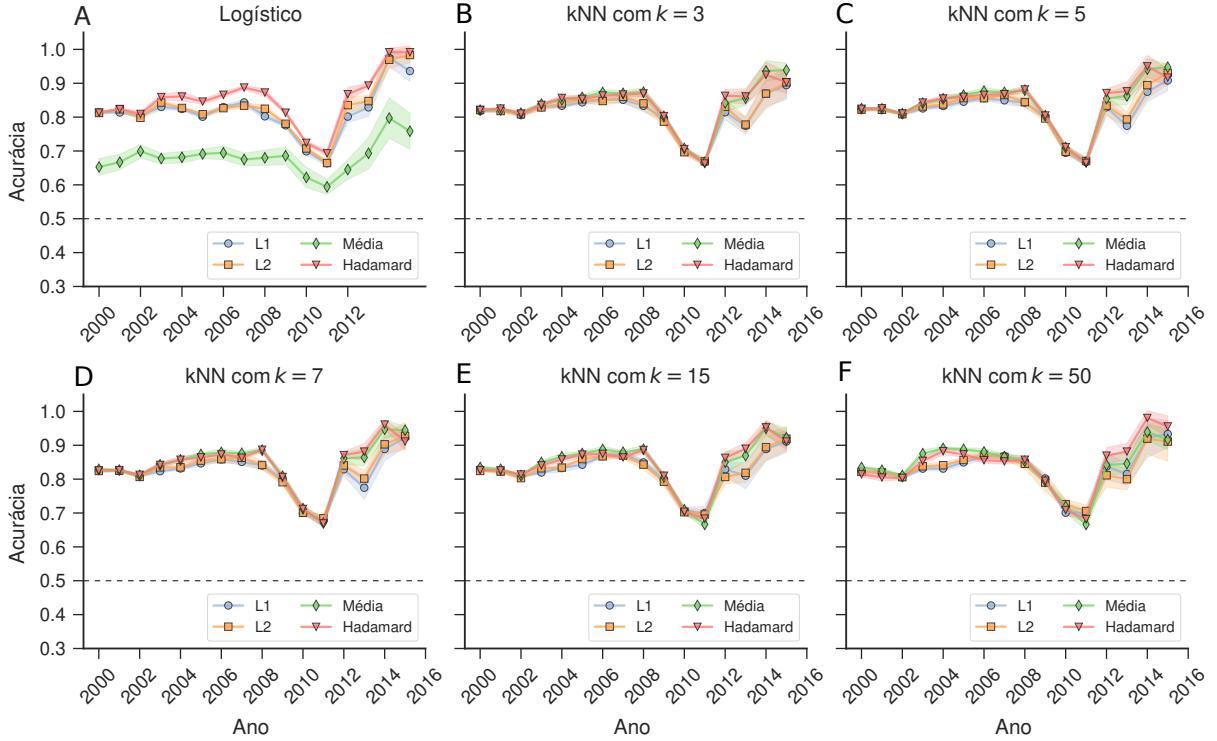


Figura 2.7: Acurácia média da previsão de parcerias futuras na rede de corrupção espanhola em função do ano limiar para diferentes operadores binários. O painel (A) mostra os resultados obtidos usando classificadores logísticos, enquanto os outros painéis [(B), (C), (D) e (E)] mostram a acurácia para classificadores *kNN* com diferentes números de vizinhos. Os marcadores representam a acurácia média nos conjuntos de teste estimados a partir de dez realizações do processo de obtenção das representações dos vértices e treinamento (as regiões sombreadas representam uma banda de desvio padrão). A linha tracejada horizontal indica a acurácia do modelo de linha de base.

Em geral, nossos resultados demonstram ser possível prever, com boa acurácia, futuros parceiros criminosos nas redes de corrupção usando apenas informações estruturais. Apesar disso, nossas acuráncias são inferiores àquelas obtidas no cenário estático (Figura 2.1A), no qual consideramos apenas o estágio final das redes para prever as ligações removidas. Assim, constatamos que prever ligações em redes que evoluem no tempo é uma tarefa mais desafiadora e os resultados obtidos em cenários estáticos podem não generalizar bem para sistemas dependentes no tempo. No próximo Capítulo, abordaremos esse mesmo problema com métodos mais sofisticados de aprendizagem de máquina e veremos que essa dificuldade ainda permanece.

2.6 Conclusões

Mostramos como a estrutura de redes criminosas pode ser usada por métodos de aprendizagem de máquina para prever ligações e variáveis relacionadas a essas ligações. Nossa pesquisa faz uso de duas redes de corrupção política, uma rede inteligência policial e uma

última rede contendo transações financeiras. Usamos o algoritmo *Node2Vec* para obter representações vetoriais de vértices (e ligações) das redes e combinamos essa abordagem com métodos de aprendizagem de máquina em uma série de tarefas preditivas.

Em um primeiro momento, lidamos apenas com as redes de corrupção e com a rede de inteligência policial. Mostramos que um classificador logístico treinado com vetores gerados via *Node2Vec* é capaz de recuperar ligações removidas dessas redes com excelente acurácia. Mais especificamente, obtemos acurárias de 98% para a rede de corrupção espanhola, 96% para a rede de corrupção brasileira e 87% para a rede de inteligência policial.

Em uma segunda aplicação, utilizamos classificadores *kNN* aplicados à rede de inteligência policial para distinguir entre relacionamentos criminosos, mistos e não criminosos. Observamos que ligações criminosas e não criminosas são classificadas corretamente em 81% e 77% das vezes, respectivamente. Por outro lado, ligações mistas são classificadas corretamente em 55% dos casos.

Em seguida, usamos um regressor *kNN* para prever as transações financeiras entre agentes na rede de lavagem de dinheiro. Observamos que o desempenho do regressor *kNN* produz valores de R^2 em torno de 0.6 e descobrimos que o número de vizinhos $k = 6$ resulta no melhor desempenho do modelo.

No quarto estudo, abordamos a tarefa de prever parcerias futuras nas redes de corrupção. Mostramos que os classificadores logísticos apresentam acurárias superiores a 0.8 para a maioria dos anos na rede de corrupção espanhola. Na rede de corrupção brasileira, notamos que as médias flutuam em torno de 0.65 e superam significativamente a acurácia do modelo de linha de base para anos posteriores a 2003. Esses resultados demonstram que a previsão de ligações em redes em que evoluem no tempo é uma tarefa mais desafiadora em comparação com cenários estáticos.

Capítulo 3

Aprendizado profundo aplicado a redes criminosas

Neste Capítulo, utilizamos redes convolucionais de grafos para realizar tarefas preditivas em redes criminosas [45]. Nossa objetivo é obter melhores resultados nas tarefas do Capítulo 2 e adicionar uma última aplicação. Para fazer isso, exploramos o *GraphSAGE*, um algoritmo de redes convolucionais de grafos. Para cada tarefa, construímos uma arquitetura de redes convolucionais composta por esses algoritmos. Além disso, empregamos uma técnica de redução de dimensionalidade para visualizar os vetores produzidos por essas arquiteturas, permitindo uma melhor compreensão do funcionamento do modelo.

Em uma primeira aplicação, realizamos a previsão de ligações nas redes. Nesse caso, lidamos com as redes de inteligência policial e com as redes de corrupção. Após obter as previsões e avaliar os resultados, estudamos como a fração de ligações no conjunto de treinamento e a dimensão dos vetores de entrada afetam a acurácia do modelo.

Em uma segunda tarefa, configuramos a arquitetura do nosso modelo para distinguir entre os tipos de associações na rede de inteligência policial. Após avaliar o desempenho do modelo, analisamos sua acurácia em relação à fração de ligações no treinamento e à variação da dimensão dos vetores de entrada.

Na terceira seção, prevemos os valores das transações na rede de lavagem de dinheiro. Avaliamos o desempenho do modelo por meio da relação entre os valores previstos e observados. Além disso, estudamos o impacto da fração de ligações no treinamento e da variação da dimensão dos vetores de entrada nas previsões do modelo.

Na quarta seção, prevemos o surgimento de futuras parcerias criminosas nas redes de corrupção. Em seguida, variamos a dimensão dos vetores de entrada e examinamos o efeito no desempenho do modelo ao longo da evolução das redes de corrupção.

Por fim, prevemos a reincidência de futuros agentes nas redes de corrupção. Diante disso, examinamos o desempenho do modelo ao longo da evolução das redes de corrupção e variamos a dimensão dos vetores de entrada para entender seu efeito na acurácia.

3.1 Prevendo parcerias em redes criminosas

Nosso foco inicial é a previsão de ligações em redes criminosas. Para fazer isso, consideramos a rede de inteligência policial e os estágios finais das redes de corrupção espanhola e brasileira. Realizamos todas as nossas tarefas preditivas por meio do algoritmo *GraphSAGE* [57]. Esse algoritmo¹ representa uma rede convolucional para grafos capaz de aprender representações vetoriais dos vértices de redes complexas. O objetivo do *GraphSAGE* consiste em gerar uma representação vetorial para cada vértice de um grafo, levando em consideração os vértices vizinhos. Os vetores resultantes desse processo podem ser empregados em diversas tarefas de aprendizagem de máquina, incluindo a previsão de ligações.

A Figura 3.1 ilustra a arquitetura do nosso modelo. Para usar o *GraphSAGE*, é necessário que todos os vértices já possuam inicialmente um vetor associado. Para gerar esses vetores iniciais, escolhemos o *Node2Vec*. Como ilustração, a Figura 3.1 mostra dois vetores de entrada (A e B) obtidos via *Node2Vec*² sendo submetidos à arquitetura do *GraphSAGE* com uma rede neural convolucional de duas camadas. A primeira camada de convolução reduz a dimensão dos vetores de entrada à metade da sua dimensão inicial, de modo que as informações concatenadas dos dois vetores tenham a mesma dimensão dos vetores de entrada. Em seguida, transferimos esse vetor concatenado para uma rede neural de duas camadas com função de ativação *ReLU*. Por fim, as informações da última camada são transmitidas para um único neurônio, o qual possui uma função de ativação *sigmoide* (correspondente à regressão logística). Esse último passo representa nossa classificação final, isto é, a previsão da presença ou ausência de ligação entre os vértices A e B³.

Para fazer previsões com essa arquitetura, dividimos os dados em conjuntos de treinamento e teste. Para criar o conjunto de treinamento, amostramos aleatoriamente 80% das ligações verdadeiras da rede e o mesmo número de ligações falsas. Por outro lado, o conjunto de teste (que nunca é usado durante o estágio de treinamento) possui 20% das ligações verdadeiras restantes e o mesmo número de ligações falsas geradas aleatoriamente.

A Figura 3.2A mostra a acurácia (fração de ligações classificadas corretamente) do modelo em função do número de épocas⁴. Para ilustrar essa análise, usamos como exemplo apenas os dados da rede de corrupção espanhola. O painel mostra duas curvas de acurácias, uma calculada a partir do conjunto de treinamento e outra a partir do conjunto de

¹O Apêndice C.14 descreve brevemente o funcionamento do *GraphSAGE*.

²Fixamos a dimensão dos vetores em 256 dimensões. Além disso, aqui e em todas as outras aplicações envolvendo o *Node2Vec* nesse Capítulo, usamos o comprimento da caminhada igual a 5, o número de caminhadas por vértice igual a 10 e o parâmetro de viés igual a 1.

³Para otimizar os parâmetros do modelo, usamos como função de custo a entropia cruzada binária descrita no Apêndice C.9. Além disso, o Apêndice C.14 apresenta detalhes sobre a arquitetura dos modelos usados nessa e nas próximas seções.

⁴Uma época significa que todos os dados do conjunto de treinamento foram usados pelo modelo uma vez e os parâmetros da rede neural foram atualizados.

teste. Mais precisamente, a cada época o modelo é treinado no conjunto de treinamento e sua acurácia é calculada nesse mesmo conjunto. Em seguida, para essa mesma época, a acurácia do modelo é calculada no conjunto de teste. Repetimos esse processo 1000 épocas. Em geral, observamos que a acurácia no conjunto de teste é sempre um pouco menor do que no conjunto de treinamento e seu valor satura, em ambos os conjuntos, após cerca de 500 épocas.

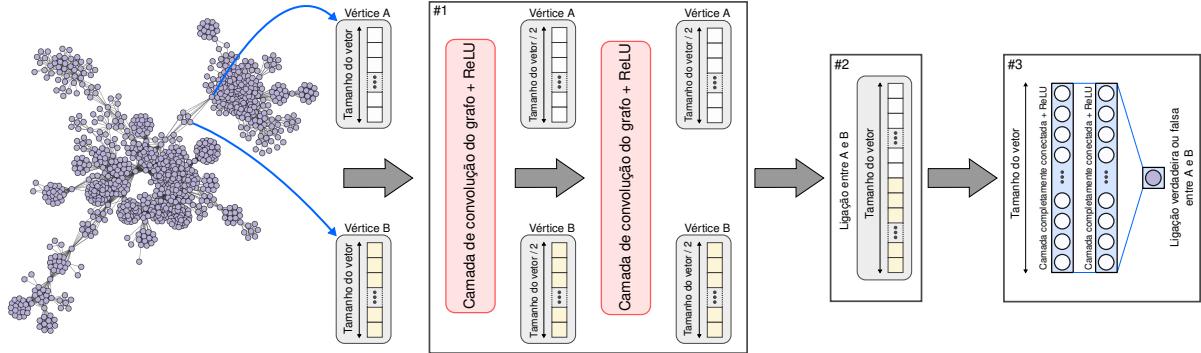


Figura 3.1: Representação esquemática da arquitetura de redes neurais usada para prever ligações nas redes criminosas. Inicialmente, cada vértice possui um vetor associado (obtido via *Node2Vec*) que passa por uma sequência de duas camadas de convolução de grafos (*GraphSAGE* com função agregadora de média) combinadas com funções de ativação *ReLU*. O vetor que sai dessas camadas de convolução possui metade da dimensão original. Em seguida, concatenamos os vetores de dois vértices para representar uma possível ligação entre esse par de vértices (A e B nesse exemplo). O vetor resultante é transmitido para uma rede neural de duas camadas. Finalmente, a classificação da ligação (representando a presença ou ausência de ligação entre os vértices A e B) ocorre na camada de saída por meio de uma função de ativação *sigmoide*.

A Figura 3.2B mostra um exemplo (usando os dados da rede de corrupção espanhola e com o modelo treinado após 1000 épocas) da matriz de confusão para a classificação de ligações verdadeiras e falsas. Nessa matriz, linhas indicam as classificações verdadeiras e colunas indicam as classificações previstas. Essa figura demonstra a eficácia do nosso modelo em distinguir entre ligações falsas e verdadeiras no conjunto de teste. Esse poder de distinção indica que a arquitetura do *GraphSAGE*, combinada com a rede neural de duas camadas, produz representações vetoriais distintas para ligações verdadeiras em comparação com aquelas geradas para ligações falsas.

Nosso próximo objetivo é entender melhor como o modelo é capaz de distinguir entre ligações verdadeiras e falsas. Para fazer isso, analisamos o vetor de 256 dimensões que aparece na última camada da rede neural. A cada previsão do modelo, esse vetor representa uma ligação verdadeira ou falsa e, portanto, vetores associados a cada uma das ligações devem possuir representações diferentes.

Uma forma de capturar essa diferença é projetar os vetores em um espaço bidimensional por meio de um método de redução de dimensionalidade. Para essa tarefa, escolhemos

o método *UMAP*⁵ devido à sua capacidade de preservar a estrutura local e global dos dados. Projetamos todos os vetores de 256 dimensões que aparecem na última camada da rede neural ao usarmos o conjunto de teste da rede de corrupção espanhola. A Figura 3.2C mostra essa projeção. Observamos que as ligações verdadeiras e falsas ocupam regiões distintas do espaço e apresentam pouca sobreposição de pontos, explicando o excelente desempenho da nossa arquitetura de redes neurais.

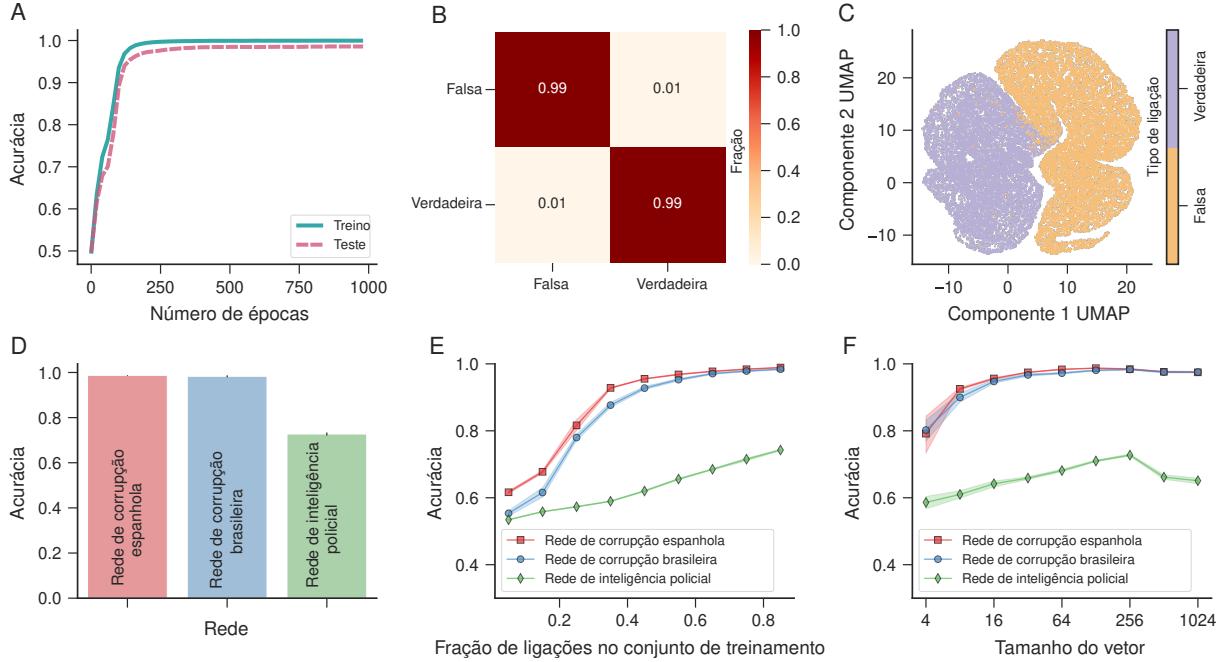


Figura 3.2: (A) Curvas de acurácia do modelo nos conjuntos de treinamento e teste em função do número de épocas utilizadas durante a fase de treinamento. (B) Exemplo de uma matriz de confusão obtida ao aplicar o modelo treinado ao conjunto de teste (as linhas indicam a classificação correta). (C) Visualização típica dos vetores da última camada do nosso modelo projetados num plano bidimensional via o método de redução de dimensionalidade *UMAP*. Nesse painel, a cor roxa indica ligações verdadeiras e a cor laranja indica ligações falsas. (D) Acurácia média das previsões do modelo nas redes de corrupção e na rede de inteligência policial. Essa média é calculada a partir de 20 realizações independentes do processo de treinamento e teste para cada rede criminosa. As pequenas barras de erro representam intervalos de confiança de 95%. (E) Acurácia média do modelo em cada rede criminosa em função da fração de ligações do conjunto de treinamento. (F) Acurácia média do modelo em cada rede criminosa em função da dimensão do vetor inicial obtido via *Node2Vec*. Nos painéis (E) e (F), os marcadores representam a acurácia média estimada a partir de 20 realizações independentes do processo de treinamento e as regiões sombreadas representam intervalos de confiança de 95%.

Até agora, tratamos dos resultados de previsões na rede de corrupção espanhola. Dando sequência ao nosso estudo, aplicamos o mesmo modelo para a rede de corrupção brasileira e para a rede de inteligência policial. Para resumir nossos resultados, calculamos

⁵O Apêndice C.13 apresenta detalhes sobre esse método. Na visualização da Figura 3.2C, usamos o número de vizinhos do *UMAP* igual a 20.

a acurácia média do modelo em distinguir entre ligações falsas e verdadeiras nas três redes criminosas. A Figura 3.2D apresenta esses valores. Nessa análise, as médias são calculadas a partir de 20 amostragens aleatórias do procedimento de divisão dos dados em treino e teste. Notamos que nosso modelo exibe um desempenho de classificação quase perfeito para ambas as redes de corrupção (99% para a espanhola e 98% para a brasileira) e uma acurácia de 73% para a rede de inteligência policial.

Uma questão interessante do processo de previsão diz respeito a como a fração de ligações no conjunto de treinamento afeta a acurácia média do modelo. Portanto, variamos a fração de ligações (de 5% a 85% com incrementos de 10%) usadas para treinar o modelo. Para cada fração, separamos os dados em conjuntos de treinamento e teste e calculamos a acurácia do modelo no conjunto de teste. Realizamos esse último procedimento 20 vezes e obtemos a média da acurácia. Conforme ilustra a Figura 3.2E, a acurácia média melhora com o aumento do número de amostras no conjunto de treinamento para as três redes criminosas. No entanto, existe um comportamento distinto entre elas. As acuráncias nas redes de corrupção se aproximam da saturação após uma porcentagem de 60% de ligações no conjunto de treinamento. Por outro lado, a acurácia na rede de inteligência policial cresce monotonicamente com o aumento da porcentagem de ligações no conjunto de treinamento.

Outra análise relevante se refere a como a variação da dimensão dos vetores produzidos pelo *Node2Vec* afeta o desempenho de nossa arquitetura. Para fazer isso, separamos os dados em conjuntos de treinamento (80%) e teste (20%) e variamos a dimensão dos vetores de entrada com incrementos espaçados logaritmicamente ($2^i \forall i \in 2, 3, \dots, 10$). Para cada uma dessas dimensões, separamos os dados em conjuntos de treinamento e teste, treinamos o modelo e obtemos sua acurácia por meio do conjunto de teste. Repetimos esse processo 20 vezes e calculamos a média da acurácia. Esse valor representa a média para uma dada dimensão. A Figura 3.2F mostra essa análise para todas as dimensões. Mais uma vez, a rede de inteligência policial apresenta um comportamento diferente das redes de corrupção. As acuráncias nas redes de corrupção se aproximam rapidamente dos valores máximos usando vetores com 256 dimensões e diminuem ligeiramente para vetores com dimensões maiores. Por outro lado, a acurácia na rede de inteligência policial aumenta aproximadamente linearmente para vetores com até 256 dimensões e diminui significativamente para dimensões maiores.

Essas diferenças, combinadas com a menor acurácia obtida para a rede de inteligência policial, indicam que aprender boas representações vetoriais para os vértices dessa rede é mais desafiador em comparação com o caso de redes de corrupção. Além disso, comparando com nossos resultados do Capítulo 2, as acuráncias observadas para as redes de corrupção são ligeiramente superiores (99% vs. 98% e 98% vs. 96% para as redes espanhola e brasileira, respectivamente), enquanto o desempenho obtido para a rede de inteligência policial é significativamente pior (73% vs. 87%). Também observamos que as curvas de

aprendizado do nosso modelo de aprendizado profundo (Figura 3.2E), em comparação com nossa abordagem anterior usando o classificador logístico (Figura 2.1E), demoram um pouco mais para atingir valores mais altos de acurácia. Isso indica que, em termos de quantidade de informações, os modelos de aprendizagem profunda tendem a ser mais custosos para treinar. Adicionalmente, é importante lembrar que a rede de inteligência policial apresenta estruturas mais complexas e menos redundantes do que as redes de corrupção, as quais são formadas por grafos completos conectados por envolvidos reincidentes (conforme discutimos no Capítulo 1). De qualquer maneira, é razoável imaginar que o desempenho do modelo seria melhor caso dispuséssemos de mais informações sobre a rede de inteligência policial.

3.2 Classificando a associação entre envolvidos da rede de inteligência policial

Prosseguindo com as tarefas de previsão, nessa seção consideramos a componente gigante da rede de inteligência policial, sobre a qual possuímos os tipos de associações entre os envolvidos (crimosa, mista e não criminosa). Nesse contexto, o objetivo é usar nossa arquitetura de redes neurais para prever qual o tipo de ligação entre dois vértices.

A Figura 3.3 apresenta a arquitetura do modelo proposto para essa tarefa. Na figura, destacamos dois vetores de entrada A e B obtidos via *Node2Vec*⁶ sendo submetidos à arquitetura de redes convolucionais do *GraphSAGE*. Esse modelo difere do anterior apenas na camada de saída, a qual é composta por três neurônios, um para cada tipo de associação. Essa camada de saída possui uma função de ativação *softmax* que normaliza os valores da distribuição de probabilidade das classes previstas. Dessa forma, a classe prevista é aquela que possui a maior probabilidade. Para otimizar os parâmetros dessa arquitetura, usamos a entropia cruzada categórica⁷ como função de custo.

Para usar essa arquitetura, separamos as ligações da rede em conjuntos de treinamento e teste. Para criar o conjunto de treinamento, amostramos aleatoriamente 80% das ligações verdadeiras da rede. O conjunto de teste, por sua vez, contém os 20% das ligações verdadeiras restantes. Nesse caso, estratificamos os dados pelas três classes, de tal forma que a proporção das associações (54% criminosas, 22% mistas e 24% não criminosas) permaneça a mesma dentro de cada conjunto.

A Figura 3.4A mostra a acurácia (fração de ligações classificadas corretamente) do modelo em função do número de épocas. O painel mostra duas curvas de acurácia, uma calculada a partir do conjunto de treinamento e outra a partir do conjunto de teste. Inicialmente, o modelo é treinado no conjunto de treinamento e sua acurácia é calculada nesse

⁶Aqui, fixamos a dimensão dos vetores em 16 dimensões.

⁷O Apêndice C.9 apresenta detalhes sobre a entropia cruzada categórica.

mesmo conjunto. Em seguida, para essa mesma época, a acurácia do modelo é calculada no conjunto de teste. Repetimos esse processo 3000 épocas. Em geral, observamos que a acurácia no conjunto de teste é sempre um pouco menor do que no conjunto de treinamento e durante todo o processo as curvas permanecem bastante próximas, indicando um ótimo desempenho do nosso modelo. Além disso, notamos que as acuráncias saturam em aproximadamente 98% após cerca de 2000 épocas.

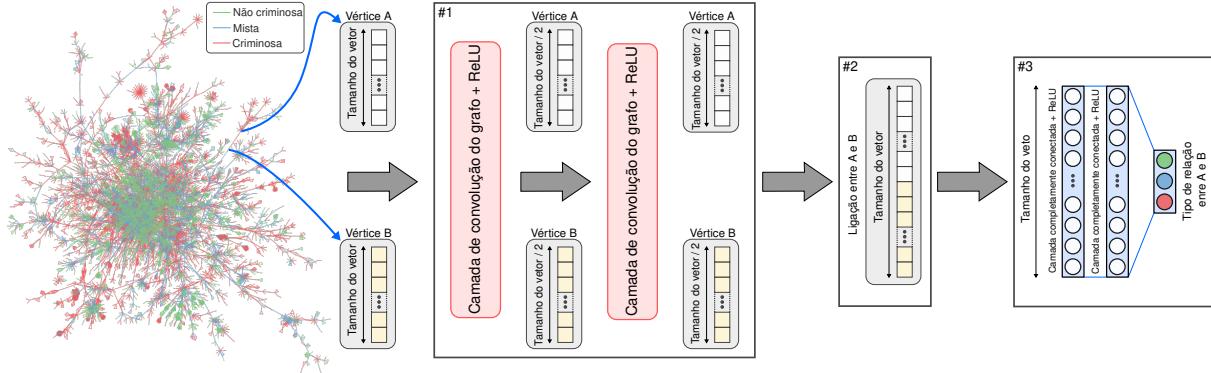


Figura 3.3: Representação esquemática da arquitetura de redes neurais utilizada para determinar o tipo de relacionamento entre envolvidos da rede de inteligência policial. Essa arquitetura é semelhante àquela usada para prever ligações. A única diferença está na camada de saída, que agora é composta por três neurônios e possui uma função de ativação *softmax*. Esses neurônios de saída representam os três tipos possíveis de relacionamento: criminoso, misto e não criminoso.

Apesar das classes das ligações serem desbalanceadas (54% criminosas, 22% mistas e 24% não criminosas), não utilizamos nenhuma estratégia para balancear a distribuição dessas classes no conjunto de treinamento. Ainda assim, conforme o resultado da matriz de confusão da Figura 3.4B, nosso modelo é bastante eficaz em distinguir entre os três tipos de associações. Nesse exemplo específico da divisão dos dados em conjuntos de treinamento e teste, o modelo classificou incorretamente apenas 3% das ligações criminosas e 3% das ligações mistas. Comparado com nossa abordagem anterior com base no classificador k -primeiros vizinhos, com a qual obtemos uma acurácia geral de aproximadamente 74% (Figura 2.3B), nosso modelo de aprendizado profundo representa uma melhoria significativa. Essa melhoria é particularmente impressionante para discriminar relacionamentos mistos, para os quais a abordagem anterior exibe uma acurácia de 55%. Mais uma vez, esse alto desempenho pode ser atribuído diretamente à qualidade das representações vetoriais dos vértices produzidas pelo modelo.

Nosso modelo é capaz de distinguir as diferentes associações com excelente acurácia. Portanto, é razoável imaginar que os vetores gerados para cada tipo de ligação ocupam regiões distintas do espaço vetorial. Para analisar esse aspecto, consideramos os vetores de 16 dimensões que aparecem na última camada da rede neural. De forma análoga à seção anterior, queremos visualizar a diferença entre os vetores ao projetá-los em um es-

paço bidimensional por meio do método de redução de dimensionalidade *UMAP*⁸. Assim, utilizamos os dados de teste e projetamos todos os vetores de 16 dimensões que aparecem na última camada da rede neural. A Figura 3.4C mostra essa projeção. Observamos que os diferentes tipos de associações ocupam regiões diferentes nesse plano. Apenas uma pequena porção de ligações mistas e criminais se sobrepõem nesta projeção bidimensional. Logo, nossos resultados confirmam a alta qualidade dos vetores produzidos pelo modelo.

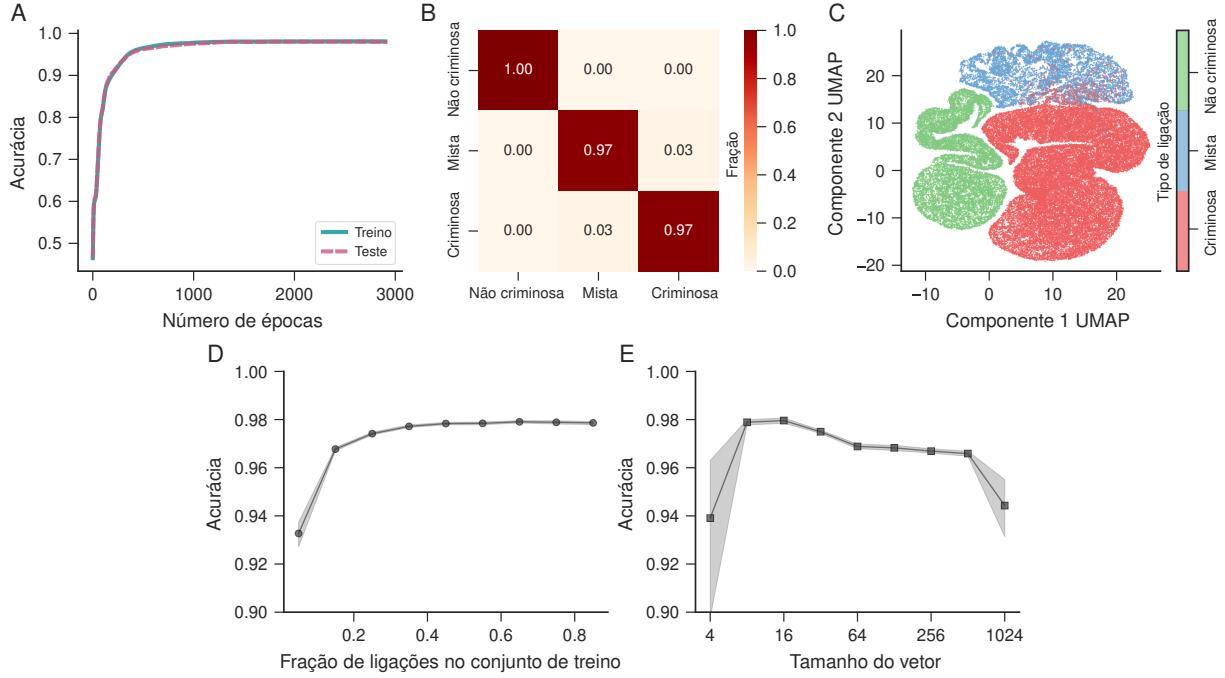


Figura 3.4: (A) Curvas de acurácia do modelo nos conjuntos de treinamento e teste em função do número de épocas utilizadas durante a fase de treinamento. (B) Exemplo de uma matriz de confusão obtida ao aplicar o modelo treinado ao conjunto de teste (as linhas indicam a classificação correta). (C) Visualização típica dos vetores da última camada do modelo projetados num plano bidimensional via o método de redução de dimensionalidade *UMAP*. Nesse painel, as diferentes cores indicam os três tipos de relacionamentos (vermelho: criminoso, azul: misto e verde: não criminoso). (D) Acurácia média do modelo sobre o conjunto de teste em função da fração de ligações do conjunto de treinamento. (E) Acurácia média do modelo sobre o conjunto de teste (com 20% de ligações) em função da dimensão do vetor inicial obtido via *Node2Vec*. Nos painéis (D) e (E), os marcadores representam a acurácia média estimada a partir de 20 realizações do processo de treinamento e as regiões sombreadas representam intervalos de confiança de 95%.

Além disso, verificamos como o desempenho do modelo muda com o número de amostras de treinamento. Variamos a fração de ligações (de 5% a 85% com incrementos de 10%) usadas para treinar o modelo e, para cada fração, separamos os dados em conjuntos de treinamento e teste e calculamos a acurácia do modelo no conjunto de teste. Realizamos esse último procedimento 20 vezes e obtemos a média da acurácia. A curva de aprendizado do nosso modelo é representada na Figura 3.4D. Observamos que a acurácia

⁸Na visualização da Figura 3.4C, usamos o número de vizinhos do *UMAP* igual a 50.

satura após considerar cerca de metade das ligações da rede nos conjuntos de treinamento. Esse comportamento também difere de nossos resultados com a abordagem do Capítulo 2 (Figura 2.3D), na qual a acurácia cresce monotonicamente com a fração de ligações nos conjuntos de treinamento.

Por fim, investigamos o papel da dimensão dos vetores gerados pelo *Node2Vec* em nossa tarefa de classificação. Assim, dividimos os dados em conjuntos de treinamento (80%) e teste (20%) e variamos a dimensão dos vetores de entrada com incrementos espaçados logaritmicamente ($2^i \forall i \in 2, 3, \dots, 10$). Para cada uma dessas dimensões, separamos os dados em conjuntos de treinamento e teste, treinamos o modelo e obtemos sua acurácia por meio do conjunto de teste. Repetimos esse processo 20 vezes e calculamos a média da acurácia. Esse valor representa a média para uma dada dimensão. A Figura 3.4E mostra essa análise para todas as dimensões. Notamos que até mesmo dimensões menores resultam em uma acurácia alta; contudo, a acurácia máxima ocorre ao utilizar vértices gerados com 16 dimensões.

3.3 Prevendo valores de transações bancárias na rede de crime financeiros

Nessa seção, lidamos com a rede de lavagem de dinheiro, na qual as ligações correspondem a transações financeiras entre os agentes. Nosso objetivo é prever o logaritmo da quantidade de dinheiro trocado entre os envolvidos. Para isso, utilizamos uma arquitetura de redes neurais baseada no *GraphSAGE* similar às abordagens das seções anteriores. No entanto, realizamos uma modificação na camada de saída para que esta possua um único neurônio com uma função de ativação linear (isto é, $f(x) = x$). A arquitetura do nosso modelo é ilustrada na Figura 3.5.

Para usar essa arquitetura, inicialmente obtemos os vetores de todos os vértices da rede via *Node2Vec*⁹. Feito isso, combinamos os vetores dos vértices que possuem ligação e separamos as ligações em conjuntos de treinamento e teste. Para criar o conjunto de treinamento, amostramos aleatoriamente 80% das ligações da rede. O conjunto de teste, por sua vez, contém os 20% das ligações restantes. Na tarefa atual, treinamos nosso modelo usando o erro quadrático médio¹⁰ como função de custo.

Para analisar nossos resultados, construímos a curva dos valores previstos versus observados. No melhor cenário, essa relação é uma reta com coeficiente angular igual a 1, o que indica que a quantidade prevista é a mesma que a observada. Para medir o desempenho do nosso modelo, calculamos o coeficiente de determinação (R^2) dessa relação. A Figura 3.6A mostra esse coeficiente para o modelo aplicado aos conjunto de treinamento

⁹Novamente fixamos a dimensão dos vetores em 16.

¹⁰O Apêndice C.8 apresenta detalhes sobre a função erro quadrático médio.

e teste em função do número de épocas de treinamento. Notamos que o coeficiente satura em aproximadamente 0.85 após cerca de 2000 épocas de treinamento.

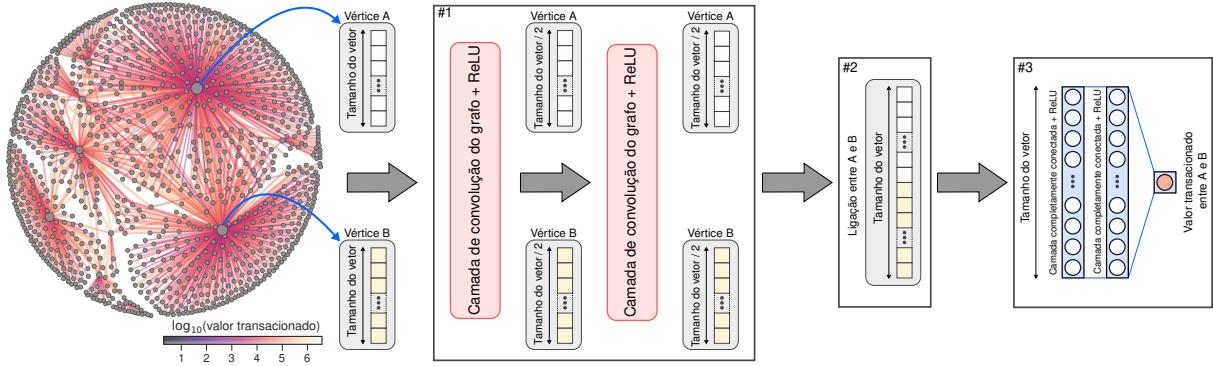


Figura 3.5: Representação esquemática da arquitetura de redes neurais usada para prever a quantidade total de dinheiro trocado entre agentes da rede de crimes financeiros. Essa arquitetura é semelhante àquelas utilizadas para prever ligações e o tipo de associação. A única diferença está na camada de saída, na qual aplicamos uma função de ativação linear para retornar o valor previsto da transação.

A Figura 3.6B apresenta a relação típica entre os valores verdadeiros e preditos no conjunto de teste utilizando o modelo treinado. Esse padrão de comportamento evidencia que o modelo produz um resultado significativamente superior em relação à nossa abordagem baseada no classificador k -primeiros vizinhos discutida no Capítulo 2, na qual obtemos um coeficiente de determinação $R^2 \approx 0.64$.

Nosso modelo possui um bom desempenho ao estimar os valores associados às ligações da rede de crimes financeiros. Para fazer essas estimativas, o modelo deve gerar, na segunda camada da rede neural, diferentes vetores conforme as diferentes ligações e seus respectivos valores. Considerando esse aspecto, realizamos uma análise análoga àquela presente nas seções anteriores, na qual visualizamos os vetores após projetá-los em um espaço bidimensional por meio do método de redução de dimensionalidade *UMAP*¹¹. Assim, projetamos todos os vetores de 16 dimensões que aparecem na última camada da rede neural ao aplicar o modelo treinado no conjunto de teste. A Figura 3.6C mostra essa projeção. Nessa figura, os pontos estão coloridos conforme o código de cores que se refere ao logaritmo da quantidade de dinheiro associada a cada ligação. Observamos que à medida que nos afastamos radialmente do centro da projeção, os valores associados às ligações tendem a diminuir, ilustrando a alta qualidade dos vetores produzidos pelo nosso modelo.

Para aprofundar nossa avaliação sobre o modelo treinado, investigamos como o coeficiente de determinação (R^2) depende da fração de ligações utilizadas durante a etapa de treinamento. Para isso, variamos a fração de ligações (de 5% a 85% com incrementos de 10%) usadas para treinar o modelo. Em seguida, calculamos o coeficiente R^2 da asso-

¹¹Na visualização da Figura 3.6C, usamos o número de vizinhos do *UMAP* igual a 200.

ciação entre os valores previstos e observados ao aplicar o modelo no conjunto de teste. Realizamos esse procedimento 20 vezes e obtemos a média da acurácia. A Figura 3.6D mostra essa relação. Nessa figura, as regiões sombreadas indicam intervalos de confiança de 95% obtidos via *bootstrap*. Observamos que o valor de R^2 se aproxima da saturação após incluirmos aproximadamente metade das ligações nos conjuntos de treinamento.

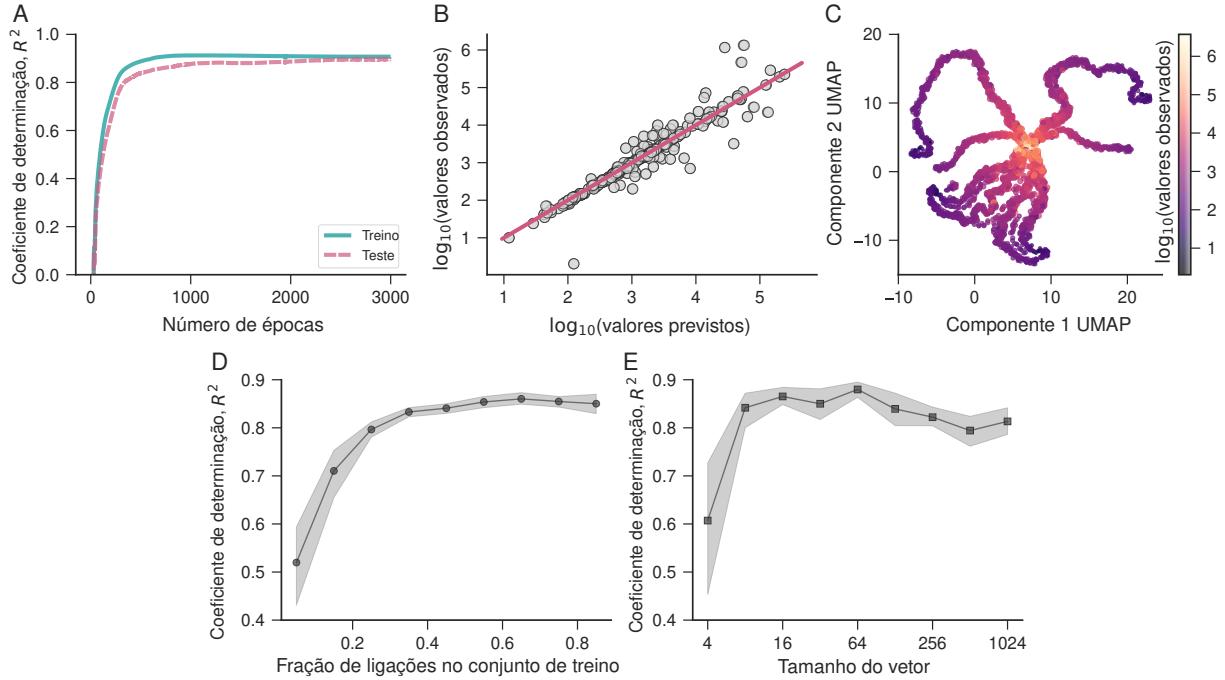


Figura 3.6: (A) Curvas do coeficiente R^2 obtidas usando o modelo aplicado aos conjuntos de treinamento e teste em função do número de épocas usadas durante a fase de treinamento. (B) Exemplo típico da relação entre os valores observados e preditos no conjunto de teste usando o modelo treinado (a linha contínua indica uma relação 1:1). (C) Visualização típica dos vetores da última camada projetados num plano bidimensional via *UMAP*. Nesse painel, o código de cores indica o logaritmo do valor da transação. (D) Valor médio do coeficiente R^2 em função da fração de ligações no conjunto de treinamento. (E) Valor médio do coeficiente R^2 em função da dimensão do vetor inicial gerado pelo *Node2Vec*. Nos painéis (D) e (E), as regiões sombreadas representam intervalos de confiança de 95% obtidos via *bootstrap*.

Para completar nosso estudo da rede de crimes financeiros, também investigamos como a dimensão dos vetores gerados pelo *Node2Vec* afeta nossa previsão em termos do coeficiente R^2 . Dessa forma, variamos a dimensão dos vetores de entrada com incrementos espaçados logaritmicamente ($2^i \forall i \in 2, 3, \dots, 10$). Para cada uma dessas dimensões, separamos os dados em conjuntos de treinamento (80%) e teste (20%), e treinamos o modelo. Em seguida, calculamos o coeficiente R^2 da associação entre os valores previstos e observados ao aplicar o modelo no conjunto de teste. Repetimos esse processo 20 vezes e calculamos a média da acurácia. Esse valor representa a média para uma dada dimensão. A Figura 3.6E mostra o valor médio do coeficiente R^2 em função da dimensão dos vetores. Conforme ilustra essa figura, notamos que as previsões são igualmente boas de 8 a 64

dimensões. No entanto, dimensões abaixo de 8 resultam em valores de R^2 consideravelmente mais baixos, enquanto dimensões acima de 64 produzem valores de R^2 ligeiramente menores.

3.4 Prevendo parcerias futuras nas redes de corrupção

Nessa seção, nosso objetivo é empregar a arquitetura de redes neurais para prever o surgimento de futuras parcerias criminosas nas redes de corrupção. Para construir os conjuntos de treinamento e teste, seguimos o mesmo procedimento detalhado na Seção 2.5 do Capítulo 2. Consideramos a rede G_Y , que engloba todos os escândalos ocorridos até o ano Y , e aplicamos o *Node2Vec* para gerar vetores de 256 dimensões associados aos vértices. No conjunto de treinamento, utilizamos todas as ligações presentes na rede e amostramos aleatoriamente o mesmo número de conexões falsas. Já para o conjunto de teste, selecionamos todas as ligações futuras entre os vértices presentes em G_Y e também amostramos aleatoriamente o mesmo número de conexões que não ocorrem no futuro.

Para fazer previsões, utilizamos uma arquitetura idêntica àquela aplicada para classificar as ligações em verdadeiras ou falsas (Seção 3.1 deste Capítulo). A Figura 3.7 exibe essa arquitetura¹². Para ilustrar o procedimento de previsão, inicialmente consideramos a rede de corrupção espanhola com todos os escândalos ocorridos até o ano de 2014. A Figura 3.8A mostra a acurácia do modelo nos conjuntos de treinamento e teste em função do número de épocas. Em ambos os conjuntos, a acurácia aumenta com o acréscimo de épocas utilizadas. Em particular, obtemos uma acurácia maior do que 90% ao usar 500 épocas.

Após treinar o modelo, estimamos a matriz de confusão usando o conjunto de teste. Os resultados da Figura 3.8B mostram que o modelo identifica corretamente todas as ligações futuras verdadeiras. Todavia, ele incorretamente classifica 7% das conexões futuras falsas como verdadeiras. No geral, verificamos que os modelos treinados são igualmente bons em discriminar entre conexões futuras verdadeiras e falsas. No entanto, esse resultado não é igual para diferentes realizações do procedimento de divisão dos dados em conjuntos de treinamento e teste. Além disso, como veremos adiante, esse desempenho também difere considerando outros estágios das redes.

¹²Na tarefa atual, treinamos o modelo usando a entropia cruzada binária como função de custo. Além disso, intensificamos os procedimentos de regularização definindo o nível de paciência para 10 épocas e o hiperparâmetro de regularização L2 para 0.002 para evitar *overfitting*.

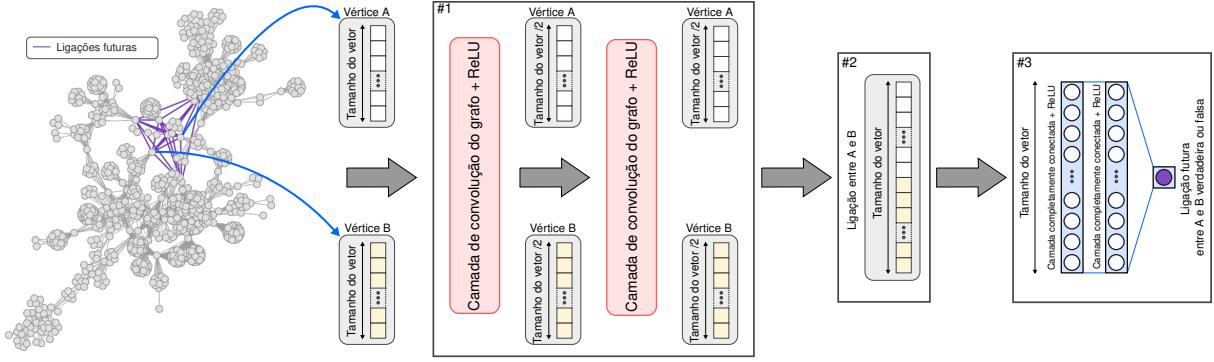


Figura 3.7: Representação esquemática da arquitetura de redes neurais usada para prever futuras ligações nas redes de corrupção. Essa arquitetura é a mesma que utilizamos para prever ligações na Seção 3.1. A rede à esquerda mostra um exemplo da rede de corrupção espanhola, no qual os vértices em roxo representam ligações verdadeiras.

Esses resultados demonstram que nosso modelo é capaz de distinguir entre ligações futuras verdadeiras e falsas. Mais uma vez, esse poder de distinção pode ser atribuído à qualidade dos vetores que a arquitetura de redes neurais produz. Portanto, investigamos os vetores de 256 dimensões que surgem na última camada do modelo ao aplicá-lo no conjunto de teste. A Figura 3.8C apresenta esses vetores projetados no plano bidimensional usando o algoritmo *UMAP*¹³. Similarmente aos padrões observados nas tarefas anteriores, notamos que as projeções das ligações verdadeiras e ligações falsas tendem a ocupar regiões distintas, apresentando pouca sobreposição.

Os resultados mostrados até agora são obtidos para uma realização do procedimento de treinamento usando a rede de corrupção espanhola com escândalos ocorridos até o ano 2014 e com vetores de entrada de 256 dimensões. Para um estudo mais abrangente, é necessário analisar o desempenho de nosso modelo ao longo da evolução das duas redes de corrupção. Em especial, estudamos esse desempenho para cada ano futuro variando a dimensão dos vetores de entrada. Examinamos as previsões do ano 2000 até 2015 para a rede espanhola e do ano 2000 até 2013 para a rede brasileira. Definimos esse intervalo de tempo a fim de garantir dados suficientes para o treinamento do modelo. Para cada um desses anos, repetimos o mesmo processo utilizado anteriormente para o ano de 2014 na rede espanhola.

Nesse contexto, variamos a dimensão dos vetores de entrada com incrementos espaçados logaritmicamente ($2^i \forall i \in 2, 3, \dots, 10$). Para cada par de ano e dimensão dos vetores, geramos os conjuntos de treinamento e teste (ambos com amostragem aleatória) conforme a mesma configuração detalhada anteriormente. Treinamos o modelo usando o conjunto de treinamento e calculamos sua acurácia no conjunto de teste. Repetimos esse processo 20 vezes e depois calculamos a média da acurácia. Essas médias estão apresentadas nas matrizes das Figuras 3.8D e 3.8E para as redes de corrupção espanhola e brasileira, respec-

¹³Na visualização da Figura 3.8C, usamos o número de vizinhos do *UMAP* igual a 20.

tivamente. Cada elemento corresponde à acurácia média para um determinado ano e uma determinada dimensão do vetor de entrada. O código de cores está associado aos valores de acurácia, enquanto valores não significativos estão indicados por elementos cinzas.

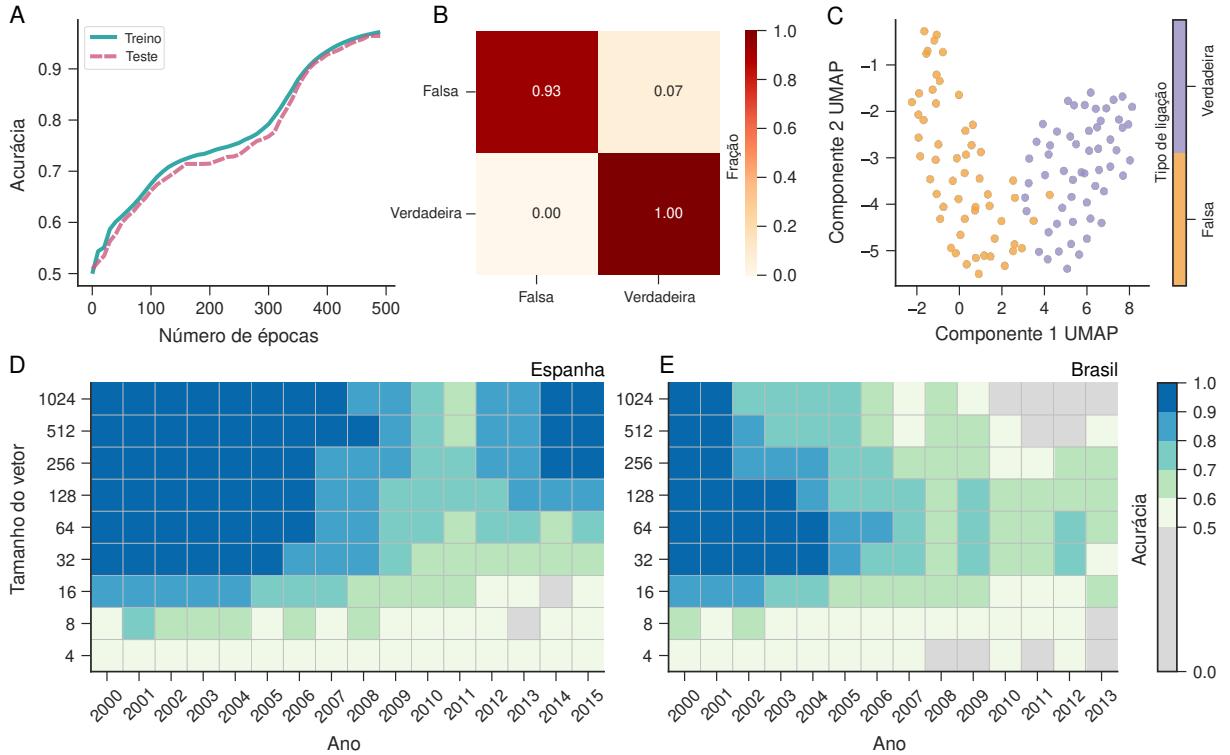


Figura 3.8: (A) Curvas de acurácia do modelo nos conjuntos de treinamento e teste em função do número de épocas utilizadas durante a fase de treinamento. (B) Exemplo de uma matriz de confusão obtida ao aplicar o modelo treinado ao conjunto de teste (as linhas indicam a classificação correta). (C) Visualização típica dos vetores da última camada do modelo projetados no plano bidimensional via *UMAP*. Nesse painel, ligações verdadeiras estão indicadas pela cor roxa e ligações falsas estão indicadas pela cor laranja. (D) Acurácia média do modelo nos conjuntos de teste da rede espanhola considerando diferentes anos e dimensões dos vetores de entrada. (E) Acurácia média do modelo nos conjuntos de teste da rede brasileira considerando diferentes anos e dimensões dos vetores de entrada. Cada elemento dessas matrizes indica a acurácia média do modelo no conjunto de teste. Essa média é estimada a partir de vinte realizações independentes do processo de divisão dos dados em conjuntos de treinamento e teste. O código de cores no canto direito inferior exibe as cores associadas aos valores médios da acurácia, com a cor cinza indicando valores não significativos.

Para ambas as redes, observamos que vetores com dimensões menores do que 16 produzem modelos com acurácia próximas ao limite de significância ou até mesmo não significativas para determinados anos. No outro extremo, vetores com dimensões muito maiores também tendem a produzir acurácias abaixo do ideal, sobretudo para a rede de corrupção brasileira. No geral, obtemos as melhores acurácias com vetores de 256 dimensões para a rede espanhola e de 64 dimensões para a rede brasileira. Essas acurácias são significativamente maiores do que aquelas obtidas com nossa abordagem anterior baseada no

classificador logístico (Figura 2.5 do Capítulo 2). Mais especificamente, para a rede espanhola, em comparação com a acurácia média de 80% obtida anteriormente (Figura 2.5 do Capítulo 2), obtemos uma acurácia média ao longo dos anos de aproximadamente 90%. Já para a rede brasileira, em comparação com a acurácia média de 65% obtida anteriormente (Figura 2.5 do Capítulo 2), obtemos uma acurácia geral de aproximadamente 80%.

Considerando a rede de corrupção espanhola, notamos que a acurácia passa por um mínimo entre 2009 e 2012. Por outro lado, para a rede brasileira, a acurácia passa por um mínimo entre 2005 e 2009. Vale ressaltar que obtivemos um comportamento parecido para a acurácia durante esses mesmos períodos em nossa abordagem do Capítulo 2 (Figura 2.5). Esse resultado reflete a coalescência das componentes das redes que relatamos no Capítulo 1 (Figura 1.4). Nesse cenário, as novas ligações formadas entre os envolvidos possuem um grau maior de complexidade e, portanto, o modelo acerta menos. Apesar disso, considerando a rede espanhola, a acurácia retorna a um valor igualmente alto. Já a rede brasileira se comporta de maneira diferente e permanece com níveis de acurácia significativamente mais baixos após a transição. Embora seja difícil fornecer uma explicação precisa para essa discrepância, podemos supor que devido ao menor tamanho da rede de corrupção brasileira, o modelo não possui informações suficientes para aprender completamente os padrões das parcerias futuras entre seus agentes. Por outro lado, esses padrões podem ser menos óbvios em comparação com o caso espanhol ou pode haver uma combinação desses fatores.

3.5 Prevendo a ocorrência de envolvidos reincidentes

Até o momento, nos concentrarmos nas previsões relacionadas às conexões entre os agentes das redes criminosas. Agora, voltamos nossa atenção para a previsão de uma propriedade específica dos vértices. Mais especificamente, abordamos o problema de identificar futuros agentes reincidentes nas redes de corrupção. Conforme indicaram nossas investigações do Capítulo 1, esse tipo de vértice desempenha um papel fundamental na estrutura e evolução das redes de corrupção. Considerando essa importância, a identificação de potenciais agentes reincidentes pode ser de grande interesse para as operações de inteligência policial.

Para formular adequadamente esse problema, consideramos a rede de corrupção (espanhola ou brasileira) G_Y , na qual incluímos todos os escândalos que ocorrem até o ano Y . Primeiramente, usamos o *Node2Vec* para gerar vetores de 256 dimensões associados aos vértices dessa rede. Para produzir os conjuntos de treinamento, simplesmente armazenamos todos os agentes reincidentes e não reincidentes presentes na componente gigante de G_Y . Por outro lado, para construir os conjuntos de teste, levamos em conta todos os envolvidos da componente gigante de G_Y que se tornarão reincidentes no futuro da rede e uma amostra aleatória (do mesmo tamanho) de vértices que não se tornarão reincidentes.

É importante ressaltar que os agentes dessas últimas amostras não estão presentes nos conjuntos de treinamento.

Nas arquiteturas aplicadas às tarefas anteriores, os vetores de entrada de dois vértices são combinados e usados conjuntamente. No entanto, para o problema atual de identificação de futuros agentes reincidentes, empregamos o vetor de apenas um vértice por vez. A Figura 3.9 mostra essa arquitetura¹⁴. Primeiramente, o vetor associado ao vértice A é submetido à rede convolucional do *GraphSAGE*. Nessa rede, a primeira camada reduz a dimensão do vetor de entrada à metade de seu tamanho inicial. O resultado dessas convoluções é então enviado para uma rede neural de duas camadas. Finalmente, a informação da última camada é encaminhada para uma camada de saída contendo um único neurônio com uma função de ativação *sigmoide*, correspondente a uma regressão logística. Essa camada gera a classificação do vértice, prevendo se ele será reincidente ou não.

Para ilustrar a previsão de envolvidos reincidentes, lidamos com a componente gigante da rede de corrupção brasileira. Mais especificamente, consideramos apenas escândalos ocorridos até 2011. Essa componente está ilustrada na Figura 3.9, na qual vértices em cinza representam agentes não reincidentes, vértices em laranja indicam reincidentes passados e vértices roxos representam agentes que se tornarão reincidentes no futuro da rede.

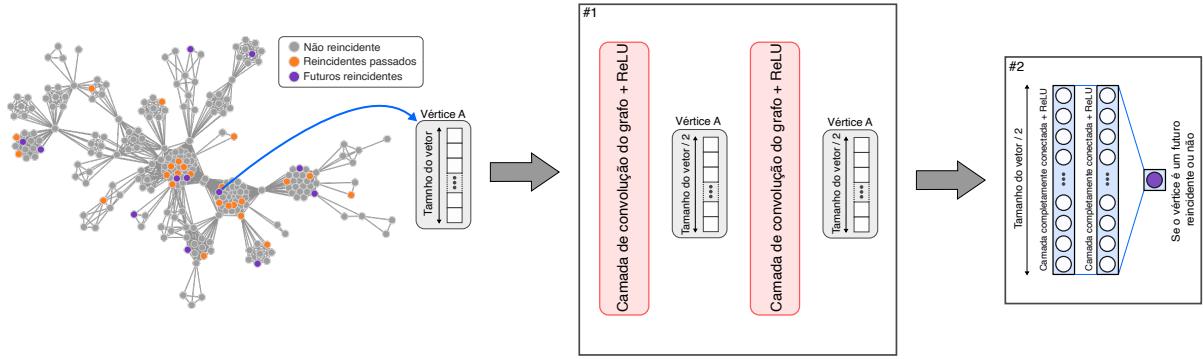


Figura 3.9: Representação esquemática da arquitetura de redes neurais usada para determinar se um agente da rede de corrupção se tornará reincidente. Essa arquitetura é semelhante àquelas aplicadas para tarefas de classificação de ligações, mas aqui não precisamos concatenar os vetores de dois vértices. Ao invés disso, usamos diretamente o vetor do vértice gerado pelo *Node2Vec* e o enviamos para a arquitetura de redes neurais. A classificação do vértice ocorre na camada de saída por meio de uma função de ativação *sigmoide*.

A Figura 3.10A apresenta a acurácia do modelo nos conjuntos de treinamento e teste em função do número de épocas. Notamos que a acurácia no conjunto de treinamento se aproxima rapidamente do valor máximo, enquanto a acurácia no conjunto de teste

¹⁴Otimizamos os parâmetros do modelo usando a entropia cruzada binária como função de custo. A essa função, aplicamos um termo de regularização L2, juntamente com seu hiperparâmetro igual a 0.001. Para mitigar o risco de *overfitting*, adotamos também um procedimento de regularização de parada antecipada com um nível de paciência de 5 épocas.

estabiliza em um nível inferior, em torno de 88%. A Figura 3.10B mostra um exemplo da matriz de confusão estimada a partir do conjunto de teste. Nesse exemplo, o modelo identifica corretamente todos os agentes não reincidentes, mas classifica erroneamente agentes reincidentes como não reincidentes 25% das vezes.

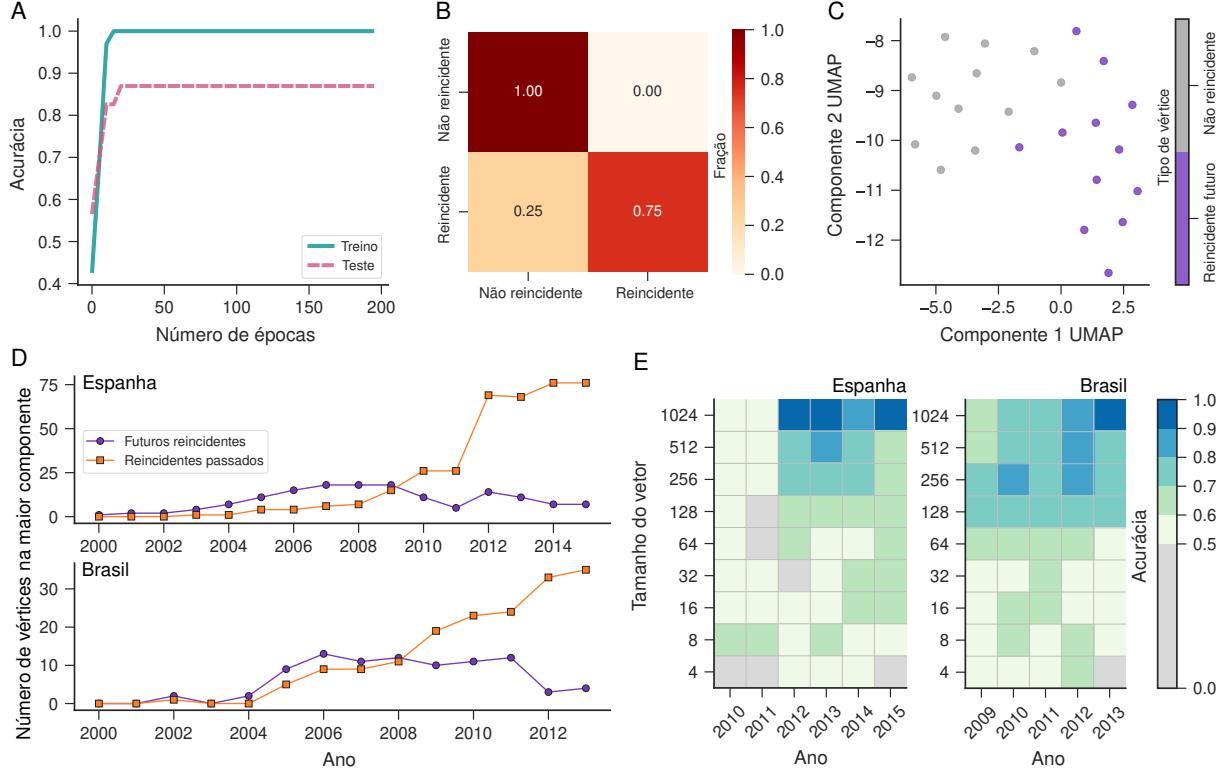


Figura 3.10: (A) Curvas de acurácia do modelo nos conjuntos de treinamento e teste em função do número de épocas utilizadas durante a fase de treinamento. (B) Exemplo de uma matriz de confusão obtida ao aplicar o modelo treinado ao conjunto de teste (as linhas indicam a classificação correta). (C) Visualização típica dos vetores da última camada do modelo projetados no plano bidimensional via *UMAP*. Nesse painel, as diferentes cores indicam se os vértices são futuros reincidentes (roxo) ou não (cinza). (D) Número de agentes reincidentes futuros e passados na componente gigante das redes de corrupção espanhola e brasileira em função do tempo. (E) Acurácia média do modelo nos conjuntos de teste ao considerar diferentes anos das redes de corrupção espanhola e brasileira, bem como diferentes dimensões dos vetores de entrada. Cada elemento dessas matrizes indica a acurácia média do modelo no conjunto de teste. Essa média é estimada a partir de vinte realizações independentes do processo de divisão dos dados em conjuntos de treinamento e teste. O código de cores no canto direito inferior exibe as cores associadas aos valores médios da acurácia, com a cor cinza indicando valores não significativos.

De maneira semelhante às tarefas anteriores, consideramos os vetores que aparecem na última camada do nosso modelo ao ajustá-lo no conjunto de teste. Nesse caso, os vetores possuem 128 dimensões, metade da dimensão inicial (256). A esses vetores, aplicamos o algoritmo *UMAP* para projetá-los em um espaço bidimensional. A Figura 3.10C exibe

essa projeção¹⁵. Observamos que (futuros agentes) reincidentes e não reincidentes tendem a ocupar regiões distintas do plano. No entanto, a separação entre as duas classes é um pouco menos clara do que aquelas observadas nas tarefas de classificação anteriores. Esse resultado se traduz nas classificações incorretas observadas na matriz de confusão (Figura 3.10B).

Até o momento, nossos resultados refletem o estudo apenas da componente gigante da rede de corrupção brasileira, com escândalos ocorridos até o ano 2011 e com vetores de entrada de 256 dimensões. Para investigar as previsões nos diferentes anos das componentes gigantes das duas redes de corrupção, primeiro calculamos a evolução do número de seus reincidentes presentes e futuros. Os resultados da Figura 3.10D mostram que leva algum tempo para observar um número não desprezível de agentes reincidentes em ambas as redes. Em especial, o número de reincidentes presentes supera o de reincidentes futuros após 2009 para a rede espanhola e após 2008 para a rede brasileira. Esse resultado é importante, uma vez que é necessário um número razoável desse tipo de envolvido para treinarmos o modelo. Portanto, para garantir amostras de treinamento suficientes, focamos nossas previsões nas redes espanhola e brasileira a partir de 2010 e 2009, respectivamente.

Para cada estágio das redes desses anos, otimizamos os parâmetros do modelo usando diferentes dimensões ($2^i \forall i \in 2, 3, \dots, 10$) dos vetores de entrada. Para cada par de ano e dimensão, geramos os conjuntos de treinamento e teste conforme o mesmo procedimento detalhado anteriormente. Treinamos o modelo no conjunto de treinamento e calculamos sua acurácia no conjunto de teste. Repetimos esse processo 20 vezes e depois obtemos a média da acurácia. As acuráncias resultantes são apresentadas nas matrizes da Figura 3.10E para as redes espanhola e brasileira. Em ambas as matrizes, observamos que as frações de classificações corretas são quase sempre inferiores a 70% para dimensões menores do que 128, com alguns casos não atingindo o nível de significância de 50%. Somente tamanhos de incorporação maiores produzem valores acima de 80%.

Por fim, verificamos que a classificação incorreta de futuros agentes reincidentes como não reincidentes (Figura 3.10B) não é uma característica incidental da rede brasileira em 2011, mas sim um padrão que observamos em todos os anos de ambas as redes. Em outras palavras, nosso modelo apresenta uma tendência de sistematicamente não identificar alguns futuros agentes reincidentes. Acreditamos que essa falha indica que a reincidência de agentes criminosos não pode ser atribuída apenas às propriedades topológicas das redes de corrupção. No nosso caso, usamos apenas informações sobre a estrutura dessas redes criminosas. No entanto, uma possibilidade a ser explorada em pesquisas futuras é verificar se a incorporação de informações adicionais, como afiliação partidária ou localização geográfica dos agentes, poderia levar a acuráncias superiores ou até mesmo revelar a existência de um caráter não determinístico na reincidência dos envolvidos.

¹⁵Na visualização da Figura 3.10C, usamos o número de vizinhos do UMAP igual a 20.

3.6 Conclusões

Ao longo desse Capítulo, apresentamos uma série de aplicações de redes neurais para grafos. Em um cenário estático, demonstramos que uma arquitetura de redes neurais pode prever parcerias criminosas em redes de corrupção e de inteligência policial. Essa arquitetura pode ser facilmente adaptada para diferentes tarefas. Por exemplo, mostramos que uma arquitetura semelhante é capaz de distinguir entre diferentes associações de envolvidos na rede de inteligência policial. Além disso, utilizamos a arquitetura de redes neurais para abordar um problema de regressão e alcançamos um desempenho alto ao prever a quantidade de dinheiro trocado entre agentes de uma rede de lavagem de dinheiro.

Em cenários dinâmicos, conseguimos antecipar propriedades das redes de corrupção usando nossa arquitetura. Demonstramos que é possível prever, com acurácia significativa, futuras parcerias entre agentes presentes nessas redes. Por fim, obtemos bons resultados ao usar o modelo para identificar futuros agentes reincidientes nas redes de corrupção.

Em todas as tarefas preditivas, verificamos que a qualidade de nossas previsões pode ser diretamente atribuída à qualidade dos vetores gerados por nossos modelos. Usando uma técnica de redução de dimensionalidade, mostramos que vetores associados a diferentes classes (em tarefas de classificação) ou associados a diferentes valores contínuos (na tarefa de regressão) tendem a ocupar regiões distintas. Dessa forma, a arquitetura de redes neurais consegue capturar essas diferenças.

Em geral, nossa abordagem supera significativamente os modelos logístico e de k -primeiros vizinhos do Capítulo 2. Observamos uma melhora de 33% na acurácia de prever o tipo de associação entre agentes da rede de inteligência policial (de 74% para 99%). Também obtemos avanços na tarefa de antecipar futuras parcerias das redes de corrupção (acuráncias com aumento de 80% para 90% na rede espanhola e de 65% para 80% na rede brasileira). Além disso, alcançamos uma melhora de 40% na tarefa de prever a quantidade de dinheiro trocado entre agentes da rede financeira (R^2 ajustado de 0.64 para 0.90). No entanto, considerando a tarefa de previsão de ligações no cenário estático, obtemos acuráncias apenas ligeiramente superiores (99% vs. 98% e 98% vs. 96% para as redes espanhola e brasileira, respectivamente), além de um desempenho significativamente pior (73% vs. 87%) para a rede de inteligência policial.

Conclusões e perspectivas

Nossa pesquisa demonstra que a estrutura de redes criminosas fornece informações relevantes sobre a associação entre criminosos. No Capítulo 1, investigamos redes de corrupção de dois países distintos e, portanto, esse estudo representa uma perspectiva interessante sobre possíveis padrões universais de estrutura e dinâmica de redes de corrupção. No entanto, nossos resultados são baseados em escândalos de corrupção de dois países ocidentais e, apesar das dificuldades em encontrar informações sobre processos de corrupção, trabalhos futuros devem ser dedicados a outros países, a fim de solidificar ou limitar os resultados que relatamos. Além disso, a falta de concordância quantitativa entre nosso modelo e algumas propriedades empíricas das redes de corrupção sugere que outros fatores, para além da reincidência criminosa, podem afetar a estrutura de redes de corrupção política. Portanto, certamente há espaço para o desenvolvimento de outros modelos, provavelmente mais complexos, para descrever o crime organizado.

Revelamos que a aplicação de algoritmos de aprendizagem estatística representa uma abordagem promissora para a análise de redes criminosas. Os resultados indicam que é possível alcançar boas acurárias na previsão de variáveis (discretas e contínuas) relacionadas a essas redes, utilizando apenas informações estruturais. Ao usar os vetores associados aos vértices, os algoritmos conseguem capturar padrões complexos de relacionamento entre os membros dessas redes. A alta acurácia e a simplicidade da implementação de métodos de aprendizagem de máquina nos permitem concluir que nossa abordagem pode ser útil em futuras operações de inteligência.

Além disso, evidenciamos o imenso potencial dos modelos de aprendizado profundo na exploração, previsão e classificação de propriedades de redes criminosas. Os modelos utilizados não apenas fornecem melhorias significativas em relação às abordagens baseadas em métodos mais simples, mas também podem inspirar novas pesquisas nessa área. Com a crescente complexidade das atividades criminosas, as aplicações de tais modelos podem auxiliar agências de aplicação da lei em suas investigações, fornecendo informações e orientações valiosas.

Contudo, é importante reconhecer as limitações do nosso trabalho. Uma delas é, sem dúvida, a qualidade das informações utilizadas para formar as redes criminosas. Apesar dos esforços para tornar essas informações confiáveis, devemos lembrar que esses dados vêm de investigações policiais de atividades ilegais e ocultas, de modo que relacionamen-

tos ausentes ou efeitos de ruído provavelmente estão presentes e afetam o desempenho de nossos métodos de aprendizagem de máquina. Esse problema também pode explicar parcialmente o desempenho inferior que observamos ao prever futuras associações criminosas. No entanto, esse tipo de problema é inerente a trabalhos empíricos relacionados a sistemas sociais.

Outra limitação do nosso trabalho é a falta de interpretações diretas de métodos de aprendizagem de máquina e a consequente dificuldade em derivar relações causais desses modelos [58–60]. Felizmente, há um consenso crescente de que, além de fornecer alta acurácia de previsão, os métodos de aprendizagem de máquina também devem ser capazes de produzir conhecimento a partir de dados, um domínio conhecido como “aprendizagem de máquina interpretável” e que está passando por rápidos desenvolvimentos [61], particularmente no contexto de aprendizagem de representação de grafos [62, 63].

Apesar dessas limitações, nossos resultados deixam claro que parcerias entre criminosos estão longe de apresentarem comportamentos completamente aleatórios. De fato, podemos afirmar que, semelhante às evidências encontradas em cenas de crime, as associações criminosas exibem padrões e carregam informações cruciais que podem ser aprendidas por métodos de aprendizagem de máquina e usadas para prever informações ausentes ou até mesmo antecipar o comportamento futuro de agentes em redes criminosas.

Apêndice A

Conceitos de ciência de redes

A.1 Definição de redes

Uma rede (ou grafo) é uma representação abstrata de objetos (vértices) e suas relações (ligações) [47]. Podemos descrever uma rede por um par (V, E) , com V sendo o conjunto de vértices e E o conjunto das ligações que conectam os vértices. Designamos a letra N para representar o número total de vértices e a letra L para caracterizar o número de ligações entre esses vértices.

Grafos podem ser direcionados (representando relações unidirecionais entre os vértices), para os quais a existência de um caminho de i para j não implica que exista um caminho de j para i . Por outro lado, grafos mais simples possuem ligações não direcionadas (relações bidirecionais) e ambos os caminhos (de i para j e de j para i) sempre existem simultaneamente. Em nosso trabalho, utilizamos apenas grafos não direcionados devido à natureza dos dados. Entretanto, na aplicação relacionada à rede de crimes financeiros, utilizamos uma rede com ligações ponderadas para representar a quantidade de dinheiro transacionada entre os agentes. Nesse caso, existe um valor associado a ligação da rede.

Uma outra representação matemática muito utilizada em teoria de grafos é a matriz de adjacência. A matriz de adjacência de uma rede com N vértices possui N linhas e N colunas, sendo seus elementos:

- $A_{ij} = 1$, caso exista ligação entre os vértices i e j .
- $A_{ij} = 0$, caso não exista ligação entre os vértices i e j .

No caso de uma rede ponderada, temos $A_{ij} = w_{ij}$, com w_{ij} indicando o peso da ligação entre os vértices i e j .

A.2 Grau e distribuição de grau

A.2.1 Grau

A partir da matriz de adjacência, podemos obter uma das grandezas mais básicas de redes, a centralidade de grau [47]. No caso mais simples, o grau de um determinado vértice representa seu número de ligações. Para uma rede de tamanho N , o grau k_i de um vértice i é calculado da seguinte maneira:

$$k_i = \sum_{j=1}^N A_{ij}, \quad (\text{A.1})$$

na qual A_{ij} são os elementos da matriz de adjacência da rede.

A.2.2 Distribuição de grau

A distribuição de grau de uma rede representa a fração de vértices que possuem um número de ligações igual a k . Para uma rede com N vértices, a probabilidade de encontrar um vértice com grau k é dada por $P(k) = \frac{N_k}{N}$, sendo N_k o número de vértices com grau k .

A maneira que escolhemos para representar a distribuição de grau nesse trabalho é por meio da distribuição acumulada complementar. Nessa abordagem, consideramos a fração de vértices que possuem grau maior do que um valor k . O Apêndice B.1 descreve em maior detalhe as distribuições acumulada e acumulada complementar, usando como exemplo a distribuição exponencial.

A.3 Medidas estruturais

A.3.1 Densidade

A densidade de um grafo [47] representa a razão entre o seu número de ligações (L) e o número máximo de possíveis ligações. Para uma rede com N vértices, o número máximo de ligações que podem ser formadas é $N(N - 1)/2$. Portanto, a densidade é definida como

$$d = \frac{L}{N(N - 1)/2}. \quad (\text{A.2})$$

A.3.2 Coeficiente de agrupamento

O coeficiente de agrupamento local [47] captura o quanto conectados são os vizinhos de um determinado vértice. Para um vértice i com grau k_i , o coeficiente de agrupamento é calculado como

$$C_i = \frac{2L_i}{k_i(k_i - 1)}, \quad (\text{A.3})$$

no qual L_i representa o número de ligações entre os k_i vizinhos de i . Se calcularmos a média dos coeficientes de agrupamento para uma rede de tamanho N , obtemos o coeficiente de agrupamento global da rede, definido como

$$\langle C \rangle = \frac{1}{N} \sum_{i=1}^N C_i. \quad (\text{A.4})$$

A.3.3 Assortatividade

O coeficiente de assortatividade [64] é uma medida que captura a tendência de vértices similares (em termos do grau) se conectarem. Essa medida representa o coeficiente de correlação de Pearson do grau entre pares de vértices conectados.

Esse coeficiente pode apresentar alguns valores que merecem destaque. Se $r = 0$, não existe tendência de vértices com graus similares se conectarem. Por outro lado, se $r > 0$, então os vértices tendem a se conectar com outros vértices com graus similares. Por fim, se esse coeficiente for negativo ($-1 \leq r < 0$), existe uma tendência de conexão entre vértices com graus diferentes.

A.3.4 Comprimento médio do caminho

O comprimento médio do caminho mais curto [47] de uma rede representa o valor médio dos caminhos mais curtos entre todos os seus pares de vértices. Essa medida é definida como

$$\langle d \rangle = \frac{1}{N(N-1)} \sum_{i \neq j}^N d_{i,j}, \quad (\text{A.5})$$

na qual N é o tamanho da rede e $d_{i,j}$ é a distância entre dois vértices i e j .

A.4 O algoritmo *Infomap*

O *Infomap* [65] é um método de detecção de comunidades (ou módulos) em redes que se baseia na minimização de uma função de custo conhecida como equação mapa¹ [66]. O algoritmo utiliza a dinâmica de caminhantes aleatórios e o fluxo de informação associado à rede para determinar as melhores partições. Mais especificamente, esse método aplica o código de Huffman [67] para associar códigos binários a cada vértice, cujos comprimentos dependem da frequência de visita do caminhante aleatório. Vértices mais visitados possuem códigos menores, enquanto vértices pouco visitados possuem códigos maiores.

A ideia é que, durante o caminho percorrido pelo caminhante aleatório, vértices que devem ser agrupados em um mesmo módulo aparecem em sequência por longos períodos. Isso faz sentido intuitivamente, uma vez que o caminhante aleatório tende a ficar mais

¹Do inglês, *map equation*.

tempo dentro de um módulo e raramente transita para outros os módulos. A sequência de *bits* que identifica o trajeto percorrido pelo caminhante aleatório é definida como aquela com o número mínimo de *bits* necessário para descrever o caminho. A partir dessa sequência, candidatos a módulos podem ser identificados. Para cada módulo, é definido um código de identificação e um código de saída. Isso permite que vértices diferentes recebam as mesmas identificações. Além disso, também há um código de identificação para sinalizar sempre que um caminhante aleatório entra em um módulo diferente. A Figura A1 ilustra esse procedimento.

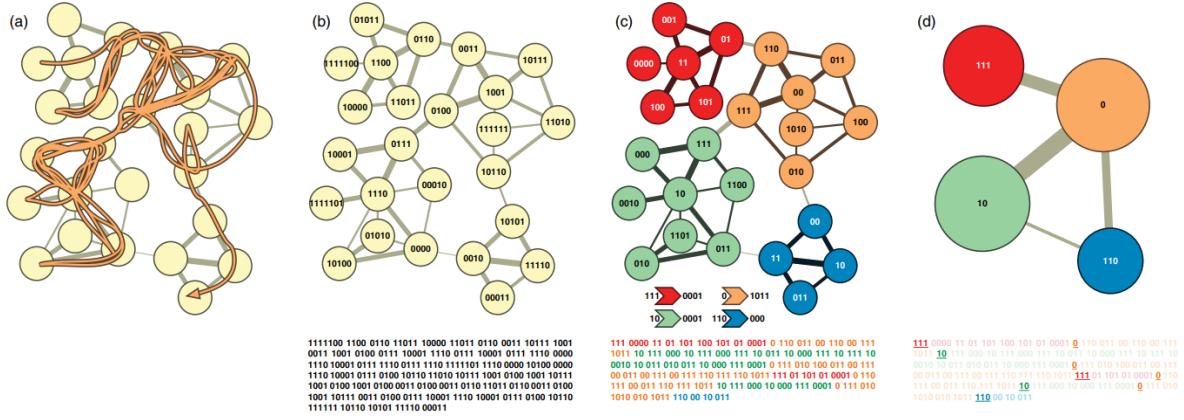


Figura A1: (a) Exemplo de uma rede percorrida por um caminhante aleatório (linha laranja). (b) Para uma descrição matemática desse percurso, os vértices recebem códigos binários segundo o código de Huffman. Os 314 bits mostrados no painel (b) descrevem a trajetória presente em (a) começando no vértice 1111100 e terminando no vértice 00011. (c) Comunidades encontradas após minimizar a equação mapa. As cores estão associadas às diferentes comunidades. (d) Rede das comunidades encontradas. Figura extraída da referência [66].

Uma vez que as possíveis partições e os caminhos foram devidamente codificados, o algoritmo tenta minimizar a equação mapa para encontrar as comunidades. Essa equação consiste em dois termos, expressados pela entropia de Shannon [68] da caminhada dentro e entre os grupos. Esses termos calculam os *bits* necessários para codificar cada um dos módulos e seus correspondentes códigos de saída, assim como os *bits* necessários para codificar cada um dos vértices dentro dos módulos. A melhor partição é aquela que minimiza o número total de *bits*, expressado na equação mapa dada por

$$L(M) = q_{\circlearrowleft} H(Q) + \sum_{i=1}^m p_{\circlearrowright}^i H(P^i), \quad (\text{A.6})$$

em que $q_{\sim} = \sum_{i=1}^m q_{i\sim}$ é a soma da probabilidade de saída para cada comunidade i e $H(Q)$ representa o comprimento médio do código do movimento entre as comunidades. No segundo termo, $p_{\circlearrowleft}^i = \sum_{\alpha \in i} p_\alpha + q_{i\sim}$ representa a probabilidade de permanência de uma

caminhada aleatória em uma comunidade i (nessa soma, p_α é a probabilidade de visitar o vértice α). Por fim, $H(P^i)$ representa o comprimento médio do código do módulo i . Portanto, o primeiro termo representa a entropia do movimento entre os módulos e o segundo termo representa a entropia do movimento dentro dos módulos.

O procedimento para minimizar a equação mapa está detalhado na referência [66]. A princípio, cada vértice representa uma comunidade única e, em cada iteração, os vértices são movidos para as comunidades que mais decrescem $L(M)$. Se nenhum movimento resultar em decréscimo da função, o vértice permanece na comunidade original. Este processo é repetido até que não haja mais decréscimo na função. Em seguida, a rede é reconstruída de forma que os módulos encontrados são os novos vértices da rede. Isso forma uma rede mais compacta, tornando-se cada vez menor a cada nível. Por esse motivo, o *Infomap* também é considerado um método hierárquico (ou multinível), uma vez que permite a detecção de comunidades dentro de comunidades. No nosso trabalho, usamos uma implementação [69] desse algoritmo disponível na linguagem de programação Python.

Apêndice B

Conceitos de estatística

B.1 Distribuição acumulada

Para ilustrar os conceitos de distribuições acumulada e acumulada complementar, consideremos uma distribuição exponencial $P(k)$ da forma

$$P(k) = \lambda e^{-k\lambda}, \quad (\text{B.1})$$

na qual λ representa o inverso do valor médio da distribuição. Ou seja, se

$$\langle k \rangle = \int_0^\infty k P(k) dk = \frac{1}{\lambda}, \quad (\text{B.2})$$

então $\lambda = 1/\langle k \rangle$.

Uma vez que definimos a distribuição exponencial e seus termos, podemos calcular sua distribuição acumulada $P(k \leq x)$. Essa função representa a probabilidade de encontrar um valor k tal que $k \leq x$. Matematicamente, escrevemos

$$P(k \leq x) = \int_0^x P(k) dk = 1 - e^{-x\lambda}. \quad (\text{B.3})$$

Nosso trabalho também faz uso da distribuição acumulada complementar, a qual pode ser escrita como

$$F(x) = 1 - P(k \leq x) = e^{-x\lambda}. \quad (\text{B.4})$$

Podemos ainda calcular o logaritmo em ambos os lados da Equação B.4:

$$\ln F(x) = -x\lambda. \quad (\text{B.5})$$

Portanto, essa equação representa uma reta em escala mono-logarítmica. Ainda nesse contexto, se fizermos uma mudança de variável da forma $x = \frac{l}{\langle k \rangle} = \frac{l}{\lambda}$, ou seja, dividirmos

os dados pelo valor médio (hipótese exponencial), obtemos

$$\ln F(l/\langle k \rangle) = -l. \quad (\text{B.6})$$

Assim, a distribuição acumulada complementar da variável reescalada x se comporta como uma reta de inclinação -1 em escala mono-logarítmica.

B.2 Método de máxima verossimilhança aplicado à distribuição exponencial

Para ajustar a distribuição exponencial da Equação B.1 a um conjunto de dados $\{x_1, x_2, \dots, x_n\}$, podemos usar o método de máxima verossimilhança. Esse método consiste na maximização da função de verossimilhança de modo que, sob o modelo estatístico assumido, os dados observados sejam os mais prováveis. Mais especificamente, a função de verossimilhança é dada por

$$\mathcal{L}(\theta|x_1, \dots, x_n) = \prod_{i=1}^n f(x_i|\theta), \quad (\text{B.7})$$

com f representando o modelo estatístico e θ os parâmetros desse modelo. Para o caso da distribuição exponencial com parâmetro λ , essa função se torna:

$$\mathcal{L}(\lambda|x_1, \dots, x_n) = \lambda^n \exp\left(-\lambda \sum_{i=1}^n x_i\right). \quad (\text{B.8})$$

No entanto, geralmente encontramos o máximo do logaritmo dessa função. O logaritmo da Equação B.2 pode ser escrito como

$$\ln \mathcal{L}(\lambda|x_1, \dots, x_n) = n \ln \lambda - \lambda \sum_{i=1}^n x_i. \quad (\text{B.9})$$

Derivando em relação a λ e igualando essa equação a zero, obtemos

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^n x_i}, \quad (\text{B.10})$$

com $\hat{\lambda}$ sendo o parâmetro da distribuição exponencial estimado para o conjunto de dados considerado.

B.3 Método *bootstrap*

Bootstrap é um método de reamostragem usado para estimar a distribuição amostral de uma quantidade estatística qualquer [70]. O objetivo do *bootstrap* é usar os dados reamostrados para fazer inferências sobre a população, tais como estimar a média da população, desvio padrão, intervalos de confiança ou testes de hipótese. O processo pode ser repetido várias vezes para gerar um grande número de amostras de *bootstrap*, as quais podem ser usadas para estimar a distribuição da estatística de interesse. Esse é um método computacionalmente simples, não paramétrico e pode ser aplicado quando a distribuição da população subjacente é desconhecida.

Para ilustrar o procedimento, vamos obter a estatística μ e seu intervalo de confiança com base em uma amostra $X = \{x_1, x_2, \dots, x_n\}$. A partir da amostra original X , escolhemos aleatoriamente n elementos para formar um novo conjunto X_i . Realizamos esse processo múltiplas vezes e calculamos μ para cada conjunto X_i . Dessa forma, obtemos um conjunto de valores para a medida μ , ou seja,

$$\bar{\mu} = \{\mu(X_1), \mu(X_2), \dots, \mu(X_n)\}. \quad (\text{B.11})$$

Usando o conjunto $\bar{\mu}$, podemos calcular o intervalo de confiança da estatística μ . Considerando que $Q_\beta[\bar{\mu}]$ representa o β -quantil da distribuição de probabilidade dos elementos de $\bar{\mu}$, então um intervalo de confiança no nível α da estatística μ é limitado inferiormente por $(Q_{\frac{\alpha}{2}}[\bar{\mu}])$ e superiormente por $(Q_{1-\frac{\alpha}{2}}[\bar{\mu}])$.

Apêndice C

Métodos de aprendizagem estatística

Aprendizagem de máquina (ou aprendizagem estatística) é uma área que explora algoritmos capazes de aprender padrões e realizar previsões em dados [71–74]. Em geral, estamos interessados em estimar uma relação (\hat{f}) entre dois conjuntos distintos, X e Y . O conjunto $X = (x_1, x_2, \dots, x_n)$ representa as variáveis independentes (ou preditoras) e o conjunto $Y = (y_1, y_2, \dots, y_n)$ representa as variáveis dependentes (ou respostas). A relação \hat{f} é estimada usando um subconjunto dos dados, chamado de conjunto de treinamento. Depois, a qualidade desse ajuste é avaliada por meio do restante dos dados, o conjunto de teste. Nesse contexto, a relação \hat{f} é frequentemente tratada como uma caixa preta¹, pois não estamos interessados na sua forma exata, desde que suas estimativas produzam boas previsões [71].

Os problemas de aprendizagem de máquina são classificados de acordo com o tipo da variável dependente. Se a variável dependente puder ser agrupada em classes ou categorias, dizemos que essa é uma tarefa de classificação. Por outro lado, se a variável dependente assumir valores numéricos contínuos, então o problema é uma tarefa de regressão [72].

A maioria dos problemas de aprendizagem de máquina se enquadram nas categorias de aprendizagem supervisionada ou aprendizagem não supervisionada. Na aprendizagem supervisionada, o conjunto Y é conhecido e, portanto, podemos estimar \hat{f} usando as variáveis independentes (X) e dependentes (Y). Além disso, podemos avaliar a acurácia do modelo comparando os valores estimados [$\hat{f}(X)$] e observados (Y). Por outro lado, na aprendizagem não supervisionada não possuímos as variáveis dependentes e, em geral, nosso objetivo é buscar padrões nos dados do conjunto X [71]. Nesse trabalho, aplicamos algoritmos de aprendizagem supervisionada (k -primeiros vizinhos, regressão logística e redes neurais e convolucionais) para realizar tarefas de classificação e regressão.

¹Do inglês, *black box*.

C.1 Função *sigmoide*

A função *sigmoide* $\sigma(t)$ mapeia qualquer número real t para um valor entre 0 e 1 [74]. Essa função é usada em aprendizagem de máquina para modelar problemas de classificação binária e em redes neurais como função de ativação. Sua forma mais comum é a função logística, definida como:

$$\sigma(t) = \frac{1}{1 + \exp(-t)}. \quad (\text{C.1})$$

A função *sigmoide* possui uma curva em forma de “S”. À medida que t aumenta, $\sigma(t)$ se aproxima de 1 e, à medida que t diminui, $\sigma(t)$ se aproxima de 0. Além disso, essa função cruza o eixo das ordenadas em $y = 0.5$ quando $t = 0$.

C.2 Função *softmax*

A função *softmax* transforma um vetor de números reais em um vetor de probabilidades normalizado [33]. Mais especificamente, essa função calcula a exponencial do valor de cada elemento e depois o divide pela soma de todas as exponenciais. Considere um vetor \mathbf{z} de dimensão k , no qual cada elemento z_i está associado a uma determinada classe. A função *softmax* pode ser escrita da seguinte forma:

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_j^k e^{z_j}}. \quad (\text{C.2})$$

Essa função converte o vetor de entrada em um vetor contendo probabilidades, no qual cada elemento se torna a probabilidade de pertencer à classe associada à sua posição. Essa transformação é especialmente útil para tarefas de classificação que possuem mais de duas classes. Assim, a previsão é feita ao escolher a classe associada ao elemento com maior probabilidade.

C.3 Regressão logística

A regressão logística é um método usado para estimar a probabilidade da variável independente pertencer a uma classe específica [74]. Consideramos o caso da regressão logística com uma variável dependente binária e uma variável independente sendo um vetor \mathbf{x} com componentes (x_1, x_2, \dots, x_n) . A regressão logística atua como um classificador com valor de saída y que pode ser igual a 1 (a observação pertence a uma dada classe) ou 0 (a observação não pertence a essa classe). Queremos calcular a probabilidade $P(y = 1|\mathbf{x})$ de que a observação pertence a essa classe e $P(y = 0|\mathbf{x})$ de que a observação não pertence a essa classe. Para tratar desse problema, primeiro consideramos uma função linear da

forma

$$z = \mathbf{w}^T \cdot \mathbf{x} + b \quad (\text{C.3})$$

com \mathbf{w} sendo um vetor coluna com pesos que representam a importância de cada componente do vetor \mathbf{x} e b representa o viés do modelo. Essa função, no entanto, não representa uma probabilidade. Queremos modelar essa equação para que ela represente uma probabilidade e retorne valores entre 0 e 1 para qualquer \mathbf{x} . Fazemos isso usando a função logística definida na seção C.1 deste Apêndice. Assim, $p(\mathbf{x})$ pode ser escrito como

$$p(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T \cdot \mathbf{x} + b)}}. \quad (\text{C.4})$$

Para usar o método de regressão logística e estimar a probabilidade $p(\mathbf{x})$ de \mathbf{x} pertencer ou não a classe, o algoritmo precisa aprender os dois parâmetros \mathbf{w} e b usando os dados de treinamento. Para fazer isso, geralmente o método de máxima verossimilhança é usado. A ideia é fazer essa estimativa de tal maneira que $p(\mathbf{x})$ retorne probabilidades altas para todas as observações \mathbf{x} pertencentes a classe ($y = 1$) e probabilidades baixas para todas as observações \mathbf{x} não pertencentes a classe ($y = 0$). Esse procedimento pode ser encontrado com detalhes na referência [71]. Por fim, de posse dos coeficientes e da probabilidade $p(\mathbf{x})$ de que uma instância \mathbf{x} pertence à classe ($y = 1$), podemos fazer uma previsão \hat{y} conforme a seguinte condição [74]

$$\hat{y} = \begin{cases} 0 & \text{se } p(\mathbf{x}) < 0.5, \\ 1 & \text{se } p(\mathbf{x}) \geq 0.5. \end{cases} \quad (\text{C.5})$$

Embora a probabilidade 0.5 como limiar para classificação seja arbitrária, nosso trabalho utiliza esse valor conforme a implementação da regressão logística presente no pacote *scikit-learn* [75] da linguagem de programação Python.

C.4 k -primeiros vizinhos

O método de k -primeiros vizinhos (ou kNN ²) é um algoritmo usado para tarefas de classificação e regressão. Para realizar uma tarefa de classificação, esse método determina a classe de uma observação com base nos pontos de sua vizinhança. Considerando um número inteiro $k > 0$ e um ponto qualquer x_0 (no espaço euclidiano, por exemplo), o algoritmo identifica os k vizinhos mais próximos a x_0 , representados por \mathcal{N}_0 . Então, o classificador estima a probabilidade condicional de x_0 pertencer à classe j como a fração de pontos em \mathcal{N}_0 que pertencem à classe j . Essa relação é escrita como

$$\Pr(Y = j | X = x_0) = \frac{1}{k} \sum_{x_i \in \mathcal{N}_0} I(y_i = j), \quad (\text{C.6})$$

²Do inglês, *k-nearest neighbors*.

na qual $I(y_i = j)$ é o número de pontos da vizinhança de x_0 que pertencem à classe j . Finalmente, o método classifica a observação x_0 à classe com maior probabilidade [71].

Para realizar uma tarefa de regressão, o método de k -primeiros vizinhos associa um valor numérico $\hat{f}(x_0)$ a uma observação x_0 após considerar uma média numérica dos valores dos vizinhos de x_0 . Matematicamente, podemos escrever³

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in \mathcal{N}_0} y_i. \quad (\text{C.7})$$

Para fazer as previsões, escolhemos arbitrariamente um valor de k e, depois, o variamos para encontrar o número de vizinhos que melhor estima nossos dados. Em outras palavras, desejamos encontrar o número de vizinhos que produz o menor erro do modelo. Uma das formas de mensurar esse erro é por meio do coeficiente de determinação (R^2), descrito a seguir.

C.5 Acurácia

A acurácia é uma medida usada para avaliar o desempenho de modelos de aprendizagem de máquina em problemas de classificação [74, 76]. Essa medida é calculada como a razão entre o número de previsões corretas e o número total de previsões realizadas.

Para ilustrar esse conceito, consideramos o caso binário no qual lidamos com duas classes A (por exemplo, positivo para uma determinada doença) e B (negativo para essa doença). Assim, existem quatro possibilidades em relação às previsões feitas por um classificador e as classes observadas. Caso a observação seja a classe A e o classificador retornar essa mesma classe, tratamos de uma previsão do tipo verdadeiro positivo (em inglês, conhecida como TP ou *true positive*). Para essa mesma observação, se o classificador retornar a classe B, obtemos um falso negativo (FN ou *false negative*). Por outro lado, se a observação for B e o classificador retornar A, então tratamos de um falso positivo (FP ou *false positive*). Por último, caso a observação seja B e o classificador retornar B, temos o caso de um verdadeiro negativo (TN ou *true negative*). Feitas essas considerações, definimos a acurácia conforme a seguinte equação

$$\frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (\text{C.8})$$

Para um problema de classificação com múltiplas classes, a acurácia é definida de seguinte forma:

$$\frac{\text{Classificações corretas}}{\text{Todas as classificações}}. \quad (\text{C.9})$$

³É importante notar que esse método pode ser facilmente extrapolado para lidar com variáveis independentes representadas por vetores [72].

Apesar de ser uma medida amplamente utilizada e intuitiva, é importante considerar suas limitações. Em particular, a acurácia pode ser enganosa em cenários com desequilíbrio significativo nas distribuições das classes, isto é, quando algumas classes têm muito mais exemplos do que outras. Em tais casos, um modelo de classificação pode alcançar uma alta acurácia simplesmente fazendo previsões consistentes com a classe majoritária, negligenciando as classes minoritárias. Em nosso trabalho, as classes dos conjuntos de treinamento e teste foram organizadas para serem balanceadas, com exceção da Seção 3.2. Nessa seção, as classes das ligações são desbalanceadas, mas não foi necessário balanceá-las porque mesmo com diferentes porcentagens (54% criminosas, 22% mistas e 24% não criminosas), o modelo conseguiu diferenciar as classes que possuem menores proporções quase 100% das vezes (Figura 3.4B).

C.6 Matriz de Confusão

A matriz de confusão fornece uma avaliação detalhada do desempenho de um modelo de classificação, revelando informações sobre a distribuição de erros e acertos em diferentes classes [74, 76]. A Figura C1 ilustra a matriz de confusão para classificações binárias. Nessa matriz, as linhas as indicam classes observadas, enquanto as colunas indicam as classes previstas. Nesse contexto, cada elemento da matriz corresponde a uma observação e previsão, caracterizando uma das quatro possibilidades: verdadeiro negativo (TN), falso positivo (FP), falso negativo (FN) ou verdadeiro positivo (TP).

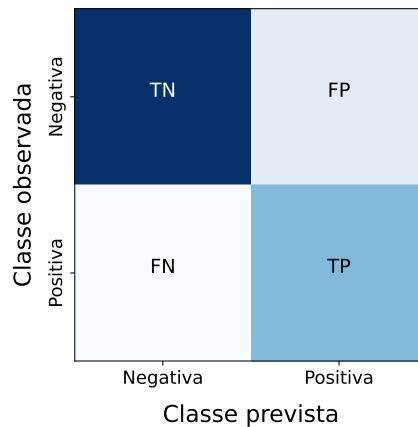


Figura C1: Matriz de confusão para classificações binárias.

A partir da matriz de confusão, é possível obter diversas medidas para o desempenho de um modelo de classificação. Um exemplo é a acurácia de um classificador binário, definida anteriormente. Nesse caso, o numerador representa a soma dos valores previstos corretamente (presentes na diagonal da matriz) e o denominador representa a soma de todas as previsões (isto é, todos os elementos da matriz).

Ainda é possível usar a matriz de confusão para apresentar previsões feitas por um modelo de classificação de múltiplas classes. A Figura C2 mostra um exemplo da matriz para esse caso.

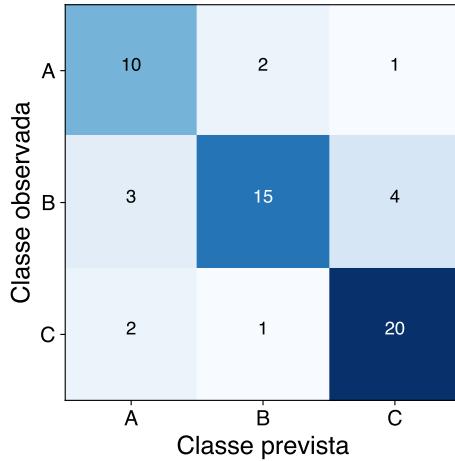


Figura C2: Exemplo da matriz de confusão para uma tarefa de classificação com múltiplas classes.

Em nosso trabalho, normalizamos os valores em todas as linhas das matrizes de confusão. Isso garante que o valor presente em cada elemento da diagonal sempre represente a fração de classificações corretamente previstas.

C.7 Coeficiente de determinação

O coeficiente de determinação (R^2) é uma medida usada para avaliar as previsões de modelos de regressão [71]. Esse coeficiente é definido como

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}, \quad (\text{C.10})$$

no qual $\text{TSS} = \sum_{i=1}^n (y_i - \bar{y})^2$ é a soma total de quadrados e $\text{RSS} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$ é a soma residual de quadrados. Nessas equações, y_i é o valor verdadeiro da variável x_i , enquanto \hat{y}_i é seu valor estimado e \bar{y} é o valor médio da variável dependente. Podemos interpretar a soma total de quadrados como a variabilidade do próprio conjunto de dados e a soma residual de quadrados como sendo a quantidade que não é explicada pelo modelo. Valores de R^2 próximos de 1 indicam que o modelo erra pouco e a regressão explica bem a dependência entre as variáveis dependente e independente. Por outro lado, valores de R^2 próximos de 0 indicam que o modelo prevê um valor de y_i sempre próximo da média y e, portanto, a regressão não captura a relação entre as variáveis⁴.

⁴É possível que R^2 assuma valores negativos, indicando que o desempenho do modelo é ainda pior.

C.8 Erro quadrático médio

O erro quadrático médio (ou MSE⁵) é outra medida para quantificar o desempenho de um modelo de regressão [71]. Se \hat{y}_i é o valor estimado para a variável x_i e seu valor observado é y_i , então sua equação pode ser escrita da seguinte forma:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (\text{C.11})$$

O resultado dessa equação depende da diferença entre os valores previstos e observados. Portanto, um erro quadrático médio menor significa que o modelo faz boas estimativas. No contexto de redes neurais, essa equação pode ser utilizada como função de custo. Nesse caso, ela também é conhecida como norma L2 (a menos do n no denominador) [74]. Na tarefa da Seção 3.3 do Capítulo 3, usamos essa medida como função de custo.

C.9 Entropia cruzada

A entropia cruzada (também conhecida como *log loss*) estima a diferença entre duas distribuições [33]. Essa medida pode ser escrita como

$$H = - \sum_y p(y) \log q(y), \quad (\text{C.12})$$

na qual $p(y)$ e $q(y)$ representam as distribuições verdadeira e estimada de y , respectivamente. No caso de tarefas de classificação de duas ou mais classes, $p(y)$ assume dois ou mais valores discretos. Para exemplificar o caso de duas variáveis dependentes, tomamos $p(y) = y$ com $y \in \{0, 1\}$. Além disso, se $q(y=1) = q$, então $q(y=0) = 1 - q$ ⁶. Portanto, reescrevemos H como⁷

$$H = -y \log(q) - (1-y) \log(1-q). \quad (\text{C.13})$$

Podemos interpretar essa equação da seguinte maneira. Quando a variável dependente for $y = 1$ [note que H se torna apenas $-\log(q)$], o melhor cenário da previsão do modelo ocorre ao estimar uma probabilidade próxima de 1 para $y = 1$ e uma probabilidade próxima a 0 para $y = 0$. Em outras palavras, $q(y=0) \rightarrow 0$ e $q(y=1) \rightarrow 1$. Considerando que $q(y=1) = q$, então q deve tender a 1. Isso faz com que $H \rightarrow 0$. Por outro lado, quando a variável dependente for $y = 0$ [note que H se torna apenas $-\log(1-q)$], o melhor desempenho do modelo ocorre quando $q(y=0) \rightarrow 1$. Considerando que $q(y=0) = 1 - q$, o valor de q deve tender a 0. Nesse caso, também observamos que $H \rightarrow 0$. Em geral, uma

⁵Do inglês, *mean squared error*.

⁶Aqui, usamos o fato de que $q(y=0) + q(y=1) = 1$, visto que $q(y)$ é uma distribuição de probabilidade.

⁷Note que a Equação C.13 é indefinida para $q = 0$ e $q = 1$.

vez que melhores previsões produzem menores valores de entropia, essa medida é utilizada como função de custo no contexto de redes neurais.

O exemplo anterior ilustra o caso da entropia cruzada binária. Podemos, ainda, tratar da entropia cruzada categórica. Para exemplificar, consideramos o caso de três classes, com $y \in \{0, 1, 2\}$ e $q(y = 0) = q_0$, $q(y = 1) = q_1$ e $q(y = 2) = q_2$. Dessa forma, podemos reescrever H como⁸

$$H = -y \log(q_0) - (1 - y) \log(q_1) - (2 - y) \log(1 - q_0 - q_1). \quad (\text{C.14})$$

Nas tarefas das Seções 3.1, 3.4 e 3.5 do Capítulo 3, usamos como função de custo a entropia cruzada binária. Nesses casos, $q(y)$ é calculado por meio da função *sigmoide*, a qual transforma os valores da camada de saída em um valor no intervalo $[0, 1]$. Já na tarefa da Seção 3.2 do Capítulo 3, usamos como função de custo a entropia cruzada categórica. Nesse contexto, $q(y)$ é obtido por meio da função *softmax*, a qual transforma o vetor da camada de saída em um vetor no qual cada elemento corresponde a uma probabilidade.

C.10 Node2Vec

Para aplicar métodos de aprendizagem de máquina em uma rede, é necessário que ela seja incorporada em uma estrutura de dados que possa ser lida por esses métodos. Essa incorporação deve ser feita de forma que o resultado preserve certas características da rede. Existem diversas técnicas para realizar essa conversão [77], incluindo o *DeepWalk* [36] e o *Node2Vec* [37].

O *DeepWalk* é um algoritmo que mapeia os vértices de uma rede em vetores. Seu objetivo é capturar a estrutura da rede por meio de vetores de uma dada dimensão, de forma que vértices semelhantes na rede possuam vetores próximos nesse espaço vetorial. A Figura C3 ilustra o objetivo principal do *DeepWalk*. Para criar esses vetores, o algoritmo gera caminhadas aleatórias na rede, as quais representam a vizinhança de um determinado vértice. Usando esses vetores, o próximo passo é treinar um modelo análogo ao *Word2Vec* [78] para obter as incorporações dos vértices.

O *Word2Vec* é um modelo de processamento de linguagem natural (PLN) que transforma conjuntos de palavras em vetores e, por meio de uma rede neural [33] de duas camadas, um de seus objetivos é detectar o contexto da palavra central conforme ela é usada (esse procedimento é conhecido como arquitetura *skip-gram*). A partir desse modelo, é possível obter a probabilidade de palavras antes e depois da palavra central. O *DeepWalk* apresenta uma generalização da modelagem de linguagem para explorar redes por meio de um fluxo de caminhadas aleatórias. A analogia é que essas caminhadas podem ser pensadas como frases, de tal forma que queremos estimar a probabilidade de observar

⁸Aqui, usamos a igualdade $q_0 + q_1 + q_2 = 1$.

o vértice v dados todos os vértices anteriores visitados pelas caminhadas aleatórias.

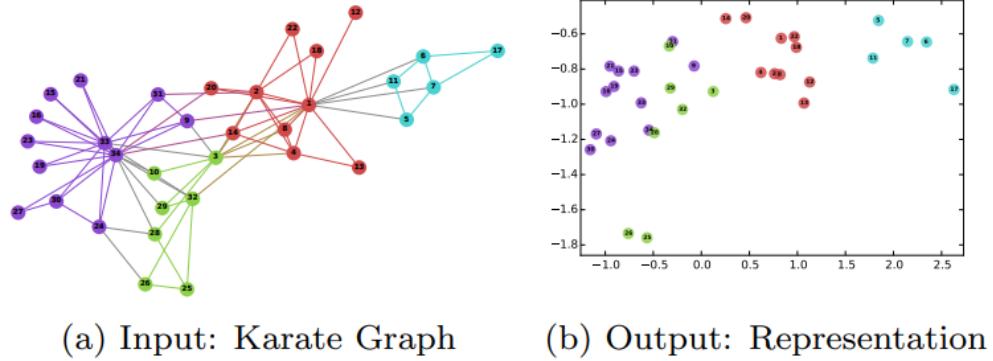


Figura C3: O objetivo do algoritmo *DeepWalk* é transformar os vértices de uma rede em vetores de uma dada dimensão. Figura extraída da referência [36].

O *Node2Vec* melhora a ideia do *DeepWalk* propondo caminhadas aleatórias tendenciosas para explorar melhor as vizinhanças da rede. Esse tipo de caminhada aleatória considera tanto o estado atual quanto o estado anterior. O processo depende de dois parâmetros (p e q) que adicionam viés a cada passo. O primeiro parâmetro (p), conhecido como parâmetro de retorno, controla a probabilidade de retornar imediatamente a um vértice que acabou de ser visitado. O segundo parâmetro (q), chamado de *in-out*, controla a probabilidade de permanecer na vizinhança de um vértice ou de visitar vértices mais distantes. O algoritmo utiliza o inverso desses valores (isto é, $1/p$ e $1/q$) e, portanto, caminhadas aleatórias com um valor alto de p têm menos chances de revisitar os vértices e promovem uma maior exploração da rede. Por outro lado, um valor alto de q faz com que o caminhante aleatório se move em direção aos vértices próximos ao vértice da etapa anterior. A Figura C4 ilustra esse processo.

A cada passo, o valor de α é definido como

$$\alpha_{pq}(t, x) = \begin{cases} 1/p, & \text{se } d_{tx} = 0 \\ 1, & \text{se } d_{tx} = 1 \\ 1/q, & \text{se } d_{tx} = 2 \end{cases} \quad (\text{C.15})$$

com d_{tx} denotando o menor caminho entre os vértices t e x .

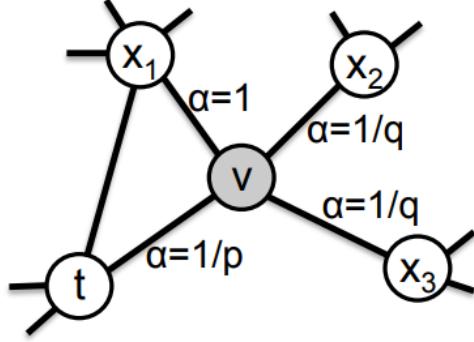


Figura C4: Um caminhante que acabou de fazer a transição de t para v avalia o próximo passo saindo de v . As legendas das ligações indicam os vieses de busca. Figura extraída da referência [37].

Usando essas caminhadas aleatórias, podemos descrever o procedimento de fazer a incorporação de vértices. Considere uma rede $G = (V, E)$ e seja $f : V \rightarrow \mathbb{R}^d$ uma função que mapeie cada vértice a um espaço vetorial de dimensão d . O objetivo é determinar f de forma que o vértice incorporado nesse espaço esteja o mais próximo possível de seus vértices vizinhos. Para fazer isso, dado um vértice de partida $u \in V$, defini-se $N_S(u) \subset V$ como uma amostra da rede de sua vizinhança (obtido a partir da caminhada aleatória). Assim, estendendo o conceito da arquitetura *skip-gram* para grafos, buscamos otimizar uma função f , a qual maximiza o logaritmo da probabilidade de observar a rede da vizinhança de um vértice u [isto é, $N_S(u)$] condicionado à função de mapeamento f . Matematicamente, isso pode ser escrito como

$$\max_f \sum_{u \in V} \log Pr(N_S(u)|f(u)). \quad (\text{C.16})$$

Em outras palavras, $Pr(N_S(u)|f(u))$ é a probabilidade de observar a vizinhança de um vértice u dada a condição de uma função (f) que mapeia esse vértice em um espaço vetorial de dimensão d . A referência [37] apresenta algumas suposições para facilitar a maximização dessa equação. Nesse trabalho, empregamos uma implementação do *Node2Vec* [79] na linguagem de programação Python.

Podemos ainda usar o *Node2Vec* para prever ligação entre dois vértices. Uma vez que as caminhadas aleatórias são naturalmente baseadas na estrutura de conectividade entre os vértices da rede, podemos estender essa ideia para pares de vértices ao agregar seus vetores. Dados dois vértices u e v , definimos um operador binário sobre os vetores $f(u)$ e $f(v)$ para gerar um vetor $g(u, v)$. Esses operadores podem ser aplicados a qualquer par de vértices, mesmo na ausência de conexão. Assim, usamos os vetores das ligações verdadeiras e das ligações falsas para treinar o modelo e realizar esse tipo de previsão.

Em nosso trabalho, consideramos quatro operadores. O operador média

$$[f(u) \boxplus f(v)]_i = (f_i(u) + f_i(v))/2, \quad (\text{C.17})$$

como o nome sugere, produz um novo vetor no qual cada elemento é o resultado da média dos elementos correspondentes dos vetores originais. O operador Hadamard

$$[f(u) \boxdot f(v)]_i = f_i(u) * f_i(v), \quad (\text{C.18})$$

gera um novo vetor no qual cada elemento é o resultado da multiplicação dos elementos correspondentes dos vetores originais. O operador $L1$

$$\|f(u) \cdot f(v)\|_{\bar{1}i} = |f_i(u) - f_i(v)|, \quad (\text{C.19})$$

produz um novo vetor no qual cada elemento é o resultado do módulo da subtração dos elementos correspondentes dos vetores originais. Por último, o operador $L2$

$$\|f(u) \cdot f(v)\|_{\bar{2}i} = |f_i(u) - f_i(v)|^2, \quad (\text{C.20})$$

gera um novo vetor no qual cada elemento é o módulo da diferença ao quadrado dos elementos correspondentes dos vetores originais.

C.11 ***LINE***

O método *LINE* (do inglês, *large-scale information network embedding*) apresenta outra abordagem para mapear vértices de uma rede em vetores [80]. Antes de explicar sua abordagem, é necessário entender a ideia de proximidade de primeira e de segunda ordem. A estrutura local de uma rede é representada pelas ligações entre os vértices, as quais caracterizam uma proximidade de primeira ordem entre eles. No entanto, a proximidade de primeira ordem em uma rede complexa não é suficiente para capturar sua estrutura global. Por isso, o algoritmo também leva em conta a proximidade de segunda ordem entre os vértices, a qual considera as conexões de suas conexões. A noção de proximidade de segunda ordem é importante porque, intuitivamente, vértices com vizinhos compartilhados possuem maior probabilidade de serem similares.

A rede da Figura C5 exemplifica essa ideia. Nessa rede, os vértices 6 e 7 estão conectados e, portanto, devem possuir uma proximidade de primeira ordem alta (esse valor pode ser quantificado pelo peso da ligação). Já entre os vértices 5 e 6 não existe ligação, mas uma vez que eles compartilham vários vizinhos, esses vértices devem possuir uma proximidade de segunda ordem alta (nesse caso, esse valor pode ser quantificado pelos pesos de seus vizinhos em comum).

Caso exista uma ligação entre dois vértices u e v , então o peso dessa ligação (w_{uv}) indica a proximidade de primeira ordem entre u e v . Se não há ligação entre esses vértices, então a proximidade de primeira ordem é zero. Por outro lado, a proximidade de segunda ordem entre dois vértices u e v é calculada pela similaridade entre seus vizinhos em comum na rede. Matematicamente, se $p_u = (w_{u,1}, \dots, w_{u,V})$ denota a proximidade de primeira ordem entre u e todos os outros vértices, então a proximidade de segunda ordem entre u e v é determinada pela similaridade entre p_u e p_v .

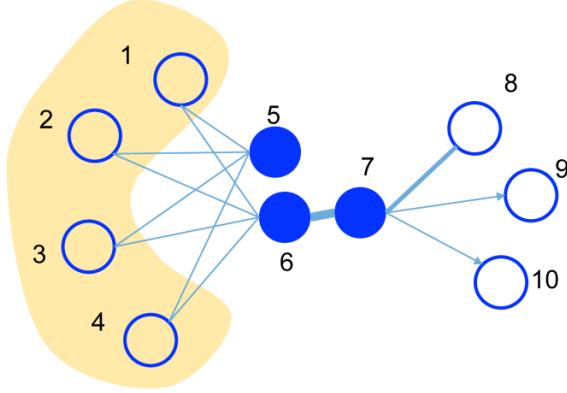


Figura C5: Uma rede simples que ilustra a ideia de proximidade de primeira e de segunda ordem entre seus vértices. Figura extraída da referência [80].

O objetivo desse método é transformar vértices em vetores preservando essas duas noções de proximidade. Para modelar a proximidade de primeira ordem, para cada ligação não direcionada (i, j) , definimos a probabilidade conjunta de dois vértices v_i e v_j como

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)}, \quad (\text{C.21})$$

na qual $\vec{u}_i \in \mathbb{R}^d$ é a representação vetorial do vértice v_i em uma dimensão d . A Equação C.21 define uma distribuição de probabilidade. Essa distribuição pode ser obtida de maneira empírica usando o peso (w_{ij}) das ligações. Nesse caso, podemos escrever

$$\hat{p}_1(i, j) = \frac{w_{ij}}{W}, \quad (\text{C.22})$$

com $W = \sum_{(i,j) \in E} w_{ij}$. Para preservar a proximidade de primeira ordem, podemos minimizar a seguinte função

$$O_1 = d(\hat{p}_1, p_1), \quad (\text{C.23})$$

na qual d representa a distância entre essas duas distribuições. Os autores utilizam a divergência de Kullback-Leibler [81] para calcular essa distância. Ao encontrar o conjunto de vetores $\{\vec{u}_i\}_{i=1,\dots,V}$ minimizando a Equação C.23, representamos cada vértice como um vetor de dimensão d . Intuitivamente, esse processo faz com que as representações vetoriais dos vértices satisfaçam à condição de proximidade de primeira ordem.

Agora podemos modelar a proximidade de segunda ordem. Para fazer isso, consideramos que cada vértice possui duas representações vetoriais associadas, \vec{u}_i e \vec{u}'_i . Então \vec{u}_i é a representação vetorial do vértice v_i e \vec{u}'_i é a representação vetorial do contexto do vértice v_i . Aqui, a palavra contexto se refere à localização do vértice na rede. Em outras palavras, cada vértice pode ser visto por sua relação direta ou indireta com outros vértices. Com esses dois vetores, para cada ligação direcionada (i, j) , podemos definir a probabilidade do vértice v_j aparecer no contexto do vértice v_i como

$$p_2(v_j|v_i) = \frac{\exp(\vec{u}'_j^T \cdot \vec{u}_i)}{\sum_{k=1}^V \exp(\vec{u}'_k^T \cdot \vec{u}_i)}. \quad (\text{C.24})$$

Para preservar a proximidade de segunda ordem, queremos que a distribuição dos contextos p_2 seja mais próxima possível da distribuição empírica \hat{p}_2 . Portanto, a ideia é minimizar a função

$$O_2 = \sum_{i \in V} \lambda_i d(\hat{p}_2, p_2), \quad (\text{C.25})$$

com d sendo a divergência de Kullback-Leibler. Os autores introduzem λ_i na equação anterior para modelar a noção de importância do vértice na rede (por exemplo, o parâmetro λ poderia representar o grau dos vértices). De forma análoga ao descrito para a proximidade de primeira ordem, a distribuição empírica $\hat{p}_2(v_j|v_i)$ pode ser escrita como

$$\hat{p}_2(v_j|v_i) = \frac{w_{ij}}{d_i}, \quad (\text{C.26})$$

na qual d_i é o grau de saída do vértice i e w_{ij} é o peso da ligação entre v_i e v_j . O procedimento e as suposições para minimizar O_1 e O_2 estão detalhados na referência [80]. Para obter os vetores a partir dos vértices, preservando tanto a proximidade de primeira ordem quanto a proximidade de segunda ordem, os autores treinam o modelo *LINE* para cada abordagem e depois concatenam as incorporações vetoriais obtidas pelos dois métodos.

C.12 Mercator

O Mercator é um método para incorporar os vértices de uma rede em coordenadas angulares por meio de um modelo geométrico [82]. Esse tipo de modelo produz redes nas quais os vértices são conectados com base em sua proximidade dentro de algum espaço métrico [83]. Um dos modelos geométricos que o Mercator considera é o modelo \mathbb{S}^1 , no qual n vértices são colocados uniformemente de forma aleatória na borda de um círculo de raio R . A cada vértice são associados dois valores: o ângulo θ do vértice nesse círculo e uma variável oculta κ proporcional ao valor esperado do grau da rede, com $\kappa \in [\kappa_0, \infty)$. Uma vez que todos os vértices estão associados a tuplas (κ, θ) , cada par de vértices se

conecta com uma probabilidade

$$p_{ij} = \frac{1}{1 + \left(\frac{d_{ij}}{\mu\kappa_i\kappa_j}\right)^\beta}, \quad (\text{C.27})$$

na qual $d_{ij} = R\Delta\theta_{ij}$ é o comprimento do arco entre os vértices i e j separados por uma distância angular $\Delta\theta_{ij}$. Os parâmetros μ e β controlam, respectivamente, o grau médio e o coeficiente de agrupamento médio da rede. A ideia é obter um conjunto de coordenadas angulares associado aos vértices $\{\theta_i, i = 1, \dots, N\}$ após estimar os parâmetros μ , β e κ_i do modelo \mathbb{S}^1 , de tal forma que os valores de grau e agrupamento médio na rede do modelo correspondam aos valores de grau e agrupamento médio observados na rede empírica. O procedimento para realizar tal tarefa é composto por diversas etapas que englobam métodos de aprendizagem de máquina, maximização de função de verossimilhança, entre outros [82].

Além disso, realizando algumas transformações algébricas, o vetor em \mathbb{S}^1 ainda pode ser expresso por um modelo geométrico no plano hiperbólico (\mathbb{H}^2). Essas são duas representações válidas (uma é a transformação da outra) do ponto de vista geométrico e, no final, podemos associar a cada vértice i da rede um vetor com duas componentes $\{\theta_i, r_i\}$. Esses modelos e os métodos para obter a representação dos vértices estão descritos em detalhes na referência [82].

C.13 UMAP

O *UMAP* (do inglês, *uniform manifold approximation and projection*) é uma técnica de redução de dimensionalidade [84]. Essa técnica é geralmente usada para visualizar dados de alta dimensão projetados em duas ou três dimensões. A partir dos dados de entrada, o algoritmo constrói uma rede geométrica e ponderada em alta dimensão e depois projeta essa rede em baixa dimensão. Esse processo deve ser realizado de forma a preservar a estrutura dos dados de entrada.

Para realizar as conexões da rede no espaço de alta dimensão, o algoritmo usa um raio para cada vértice, conectando dois vértices quando a distância entre eles for igual ou menor a r . Esse raio desempenha um papel crítico, uma vez que valores pequenos de r geram aglomerados menores e isolados, enquanto raios muito grandes geram uma rede excessivamente conectada. Para resolver esse problema, o algoritmo escolhe o raio com base na distância até o k -ésimo vizinho mais próximo de cada vértice.

No entanto, quando tratamos de um espaço de alta dimensão, nos deparamos com o problema da “maldição de dimensionalidade” [85], o qual se refere ao fato dos dados se tornarem muito dispersos em alta dimensão. Por sua vez, esse problema torna a escolha do melhor r um problema não trivial. Para resolver essa dificuldade, o *UMAP* usa

um raio variável a partir de cada vértice com base na distância até seu k -ésimo vizinho mais próximo. Assim, para cada conexão possível existe uma probabilidade, de tal modo que pontos mais longes são menos prováveis de serem conectados. Para nenhum vértice ficar isolado, cada vértice deve estar conectado pelo menos a seu vizinho mais próximo. Portanto, o algoritmo gera uma rede geométrica e ponderada em alta dimensão, na qual os pesos das ligações representam a probabilidade de dois pontos estarem conectados. A equação do peso das ligações (associado à probabilidade de conexão) entre dois vértices é dada por

$$w((x_i, x_{i_j})) = \exp\left\{\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right\}, \quad (\text{C.28})$$

com $d(x_i, x_j)$ sendo a distância entre os dois vértices, σ_i um fator de normalização e

$$\rho_i = \min\{d(x_i, x_{i_j}) \mid i \leq j \leq k, d(x_i, x_{i_j}) > 0\} \quad (\text{C.29})$$

a menor distância entre o vértice i e seu primeiro vizinho mais próximo. Dessa forma, a probabilidade de conexão diminui à medida que o raio aumenta.

Depois que a rede em alta dimensão é construída, o *UMAP* usa um algoritmo de *layout* de redes baseado em força entre os vértices para fazer a projeção em baixa dimensão. Nesse *layout*, a força de atração entre dois vértices i e j com peso de ligação $w((x_i, x_j))$ e posição $\mathbf{y}_i, \mathbf{y}_j$ é dada por

$$\frac{-2ab\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2(b-1)}}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2} w((x_i, x_j))(\mathbf{y}_i - \mathbf{y}_j). \quad (\text{C.30})$$

Por sua vez, a força de repulsão é calculada via

$$\frac{2b}{(\epsilon + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2)(1 + a\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2b})}(1 - w((x_i, x_j)))(\mathbf{y}_i - \mathbf{y}_j), \quad (\text{C.31})$$

com ϵ sendo um número pequeno para evitar a divisão por zero (0.001 na implementação usual).

No *UMAP*, essas equações de força são derivadas de uma otimização usando como função de custo a entropia cruzada entre a rede G (em alta dimensão) e uma rede equivalente H (em baixa dimensão) construída a partir dos pontos $\{\mathbf{y}_i\}_{i=1,\dots,N}$. Dessa forma, o algoritmo tenta posicionar os pontos y_i de forma que a rede pesada H se aproxime da rede original G . Essa diferença entre as redes é medida pela entropia cruzada entre as probabilidades (pesos) das ligações. Uma vez que a rede em alta dimensão captura a topologia dos dados de origem, a rede equivalente em baixa dimensão corresponde à topologia tanto quanto a otimização permitir [84]. No geral, o *UMAP* é uma maneira rápida e poderosa para analisar dados de alta dimensão por meio de uma representação mais simples e comprehensível. Usamos a implementação do *UMAP* presente na biblioteca *umap-learn* [86] da linguagem de programação Python. O algoritmo *UMAP* possui 4 parâmetros: número

de vizinhos ($n_neighbors$), distância mínima entre pontos na representação de baixa dimensão (min_dist), número de dimensões do espaço de baixa dimensão ($n_components$), e a métrica do espaço para calcular as distâncias ($metric$). Em todas as nossas análises, usamos min_dist igual a 0.99, número de dimensões igual a 2 e métrica euclidiana. Além disso, o número de vizinhos é informado em cada aplicação que empregamos o método UMAP.

C.14 GraphSAGE

O *GraphSAGE* [57] é um algoritmo iterativo capaz de aprender representações vetoriais dos vértices de um grafo⁹. Esse algoritmo representa uma rede convolucional para grafos e possui uma abordagem indutiva. Isso significa que o modelo treinado na estrutura de um grafo específico pode ser aplicado mesmo quando a estrutura do grafo for diferente, como no caso de novos vértices.

De forma geral, o *GraphSAGE* amostra um número fixo de vizinhos para cada vértice até um determinado número de saltos¹⁰ e, em seguida, usa uma função que agrupa informações dessa vizinhança e as concatena com informações do próprio vértice. Mais especificamente, se h_u^k é a representação vetorial do vértice u após a iteração k e h_u^0 é seu vetor de entrada¹¹, podemos escrever o processo iterativo do *GraphSAGE* como

$$\begin{aligned} h_{\mathcal{N}_u}^k &= \text{AGG}_k \left(\{h_v^{k-1}, \forall v \in \mathcal{N}_u\} \right) , \\ h_u^k &= \sigma \left(W^k \cdot \text{CONCAT}(h_u^k, h_{\mathcal{N}_u}^k) \right) . \end{aligned} \quad (\text{C.32})$$

Nessas equações, \mathcal{N}_u representa a vizinhança amostrada do vértice u e AGG_k é a função usada para agrregar os vetores dos vértices vizinhos de u para gerar $h_{\mathcal{N}_u}^k$. Além disso, σ representa a função de ativação *ReLU* [74], CONCAT indica uma operação de concatenação e W^k é uma matriz com pesos (parâmetros) a serem optimizados. A Equação C.32 é iterada de $k = 1$ a $k = K$, com K representando a profundidade da pesquisa ou o número máximo de saltos. Cada iteração k é muitas vezes referida como uma camada convolucional da rede, de tal forma que podemos pensar as informações agregadas h_u^k como neurônios de uma rede neural totalmente conectada. Nesse trabalho, utilizamos o operador média como função de agregação (AGG_k) em todos as nossas análises devido a sua simplicidade e desempenho em testes preliminares¹². A camada final gera uma representação vetorial (de comprimento fixo) para cada vértice, a qual pode ser usada em tarefas de regressão e classificação.

⁹Nesta seção, nos referimos a redes complexas como grafos e reservamos a palavra redes para tratar de redes neurais e redes convolucionais.

¹⁰Do inglês, *hops*.

¹¹Esse vetor de entrada pode ser obtido via *Node2Vec*, como fizemos no Capítulo 3.

¹²A referência [57] mostra algumas outras opções para a função de agregação.

A intuição geral por trás desse algoritmo é que a cada iteração os vértices agregam informações de seus vizinhos locais e, à medida que esse processo se repete, as representações vetoriais dos vértices obtêm cada vez mais informações. O motivo do *GraphSAGE* ser indutivo é justamente porque, de posse da matriz W_k com os pesos aprendidos, a representação vetorial de um novo vértice pode ser gerada a partir de sua vizinhança. Essa matriz representa uma noção da importância geral da vizinhança dos vértices de cada camada. A Figura C6 ilustra esse processo para vértices de um grafo pequeno.

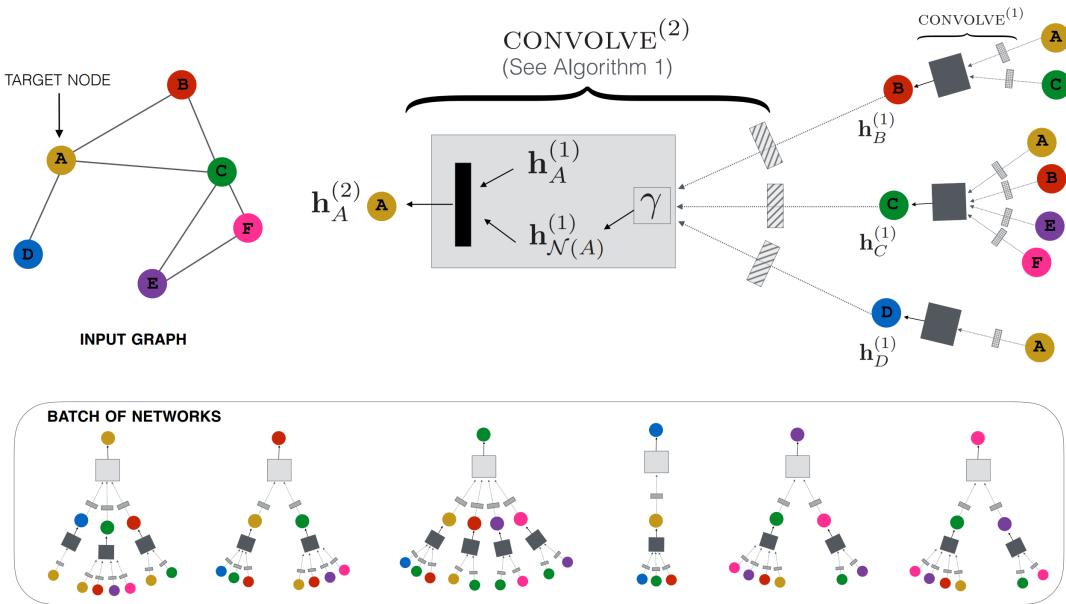


Figura C6: Visão geral da arquitetura de uma rede convolucional de grafos com profundidade $k = 2$. Esquerda: um grafo de tamanho pequeno. Direita: a rede neural de 2 camadas que calcula a incorporação $h_A^{(2)}$ do vértice A usando a representação da camada anterior $h_A^{(1)}$ do vértice A e de sua vizinhança $h_{N(A)}^{(1)}$ (vértices B, C e D). Parte inferior: redes neurais que calculam as incorporações de cada vértice do grafo de entrada. Embora as redes neurais sejam diferentes de vértice para vértice, cada camada k compartilha a mesma função de agregação (AGG_k , aqui representada por γ) e os mesmos parâmetros da matriz W^k . Nessa figura, caixas retangulares com os mesmos tons de cinza compartilham os mesmos parâmetros. Figura extraída da referência [87].

As tarefas que realizamos usando o *GraphSAGE* constituem problemas de aprendizado supervisionado. Portanto, é necessário que os parâmetros de W^k sejam aprendidos de tal forma que as previsões sejam as melhores possíveis. Para atualizar esses parâmetros conforme nosso objetivo, usamos os vetores resultantes da última camada convolucional do *GraphSAGE* como vetores de entrada em uma rede neural de duas camadas [33]. Essa última rede possui ainda uma camada de saída responsável pelas previsões, permitindo o cálculo da função de custo e a atualização de todos os parâmetros dessa configuração. Dessa forma, nosso estudo utiliza um modelo (ou arquitetura) de redes neurais, que consiste de uma combinação de redes convolucionais para grafos (*GraphSAGE*) com redes neurais. Para trabalhar com essa arquitetura, usamos a biblioteca *PyTorch* [88] da lin-

guagem de programação Python. Essa biblioteca possui implementados todos os métodos e algoritmos discutidos nesse Apêndice.

Um dos passos mais cruciais dessa arquitetura de redes neurais se refere a como atualizar os pesos das matrizes de convolução do *GraphSAGE* e da rede neural de duas camadas. O algoritmo responsável por esse processo é o método de gradiente estocástico, o qual atualiza simultaneamente os pesos da rede de duas camadas e da matriz W^k . Em especial, aplicamos o método de gradiente estocástico conhecido como Adam [89]. Esse método possui um único parâmetro, a taxa de aprendizado, que usamos igual a 0.001 para todas as tarefas do nosso trabalho. De forma geral, combinado com as funções de custo discutidas anteriormente, o esse método Adam permite aprender todos os parâmetros da arquitetura de redes neurais.

Muitas vezes, as arquiteturas são bastante complexas e apresentam uma grande tendência de memorizar as informações presentes no conjunto de teste. Isso é um problema, uma vez que o objetivo é obter um modelo com bom desempenho no conjunto de teste. Naturalmente, a rede neural precisa generalizar bem para dados relativamente diferentes. Esse problema é conhecido na literatura de aprendizagem de máquina como *overfitting* [74]. Para prevenir que nossos modelos tenham esse problema, aplicamos uma técnica de regularização [74]. Essa técnica adiciona um termo de penalidade na função de custo para diminuir a complexidade do modelo ao mudar os valores de certos pesos presentes na função de custo. Mais especificamente, aplicamos a regularização L2 [74]. Esse tipo de regularização, em especial, representa a soma dos quadrados dos parâmetros (nesse caso, da rede neural) multiplicada por uma constante (hiperparâmetro), a qual empregamos o valor de 0.001.

Em nosso trabalho também usamos um procedimento conhecido como parada antecipada¹³. A parada antecipada é outra técnica usada para evitar o *overfitting*. A ideia por trás desse método de regularização é monitorar o desempenho do modelo no conjunto de teste e interromper o treinamento quando o desempenho parar de melhorar. Isso significa que o modelo começou a superajustar os dados de treinamento e não está aprendendo padrões gerais.

Além disso, mais alguns conceitos merecem destaque. Em redes neurais, uma época se refere a uma passagem completa por todo o conjunto de dados de treinamento durante a fase de treinamento. O objetivo de treinar uma rede neural é minimizar a função de custo e, para isso, normalmente os dados de treinamento são divididos em partes menores (nós meados *batches*), de tal forma que o algoritmo de optimização (por exemplo, o gradiente estocástico) atualiza os parâmetros da rede após cada *batch*. Após todos os *batches* terem sido processados, uma época é concluída. Para utilizar a parada antecipada, é definido um parâmetro conhecido como nível de paciência. Esse parâmetro representa o número de épocas que esperamos antes de parar o treinamento caso o desempenho da rede não

¹³Do inglês, *early stopping*.

melhore.

Embora a arquitetura usada em todas as nossas previsões no Capítulo 3 seja essencialmente a mesma, cada tipo de tarefa requer alterações específicas. Na Seção 3.1 do Capítulo 3, abordamos o problema de classificação binária e prevemos se as ligações são verdadeiras (1) ou falsas (0). Mais especificamente, empregamos os vetores (obtidos via *Node2Vec*) de dois vértices quaisquer nas redes convolucionais do *GraphSAGE*. Em seguida, concatenamos os vetores resultantes desse processo, de tal forma que o vetor concatenado representa uma possível ligação. Depois, esse vetor é transferido para a rede neural de duas camadas conectadas e, na camada de saída (que consiste em um único neurônio), os valores são transformados por meio de uma função *sigmoide* (semelhante a uma regressão logística). Para treinar essa arquitetura, utilizamos a entropia cruzada binária como função de custo. Na Seção 3.2 do Capítulo 3, classificamos o tipo de associação (criminosa, mista e não criminosa) das ligações. Para fazer isso, empregamos 3 neurônios na camada de saída em combinação com a função de ativação *softmax*. Nesse caso, obtemos 3 probabilidades e a escolha é feita a partir da classe que possui maior probabilidade. Para treinar essa arquitetura, também usamos como função de custo a entropia cruzada (dessa vez, categórica). Na Seção 3.3 do Capítulo 3, abordamos uma tarefa de regressão e usamos a camada de saída com um único neurônio que retorna um valor contínuo. Nesse contexto, utilizamos uma função de ativação linear [$f(x) = x$]. Para treinar essa arquitetura, utilizamos como função de custo a função erro quadrático médio. Na Seção 3.4 do Capítulo 3, novamente tratamos de um problema de classificação binária, no qual prevemos ligações futuras nas redes de corrupção. Nessa tarefa, a arquitetura é a mesma daquela presente na Seção 3.1. Nesse caso, otimizamos os parâmetros do modelo com os mesmos procedimentos usados anteriormente e usamos a entropia cruzada binária como a função de custo. Na Seção 3.5 do Capítulo 3, classificamos se envolvidos nas redes de corrupção irão se tornar reincidentes. Portanto, tratamos de uma classificação de vértices. Usamos basicamente a mesma arquitetura das Seções 3.1 e 3.4, com a diferença de não combinar os vetores dos vértices de entrada (como na previsão de ligações). Dessa forma, o vetor de cada vértice obtido via *Node2Vec* passa pela arquitetura de redes neurais e a classificação é feita na camada de saída.

Referências Bibliográficas

- [1] Jensen, H. J. *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems* (Cambridge University Press, Cambridge, 1998).
- [2] Mitchell, M. *Complexity: A Guided Tour* (Oxford University Press, New York, 2009).
- [3] Castellano, C., Fortunato, S. & Loreto, V. Statistical physics of social dynamics. *Reviews of Modern Physics* **81**, 591, DOI: [10.1103/RevModPhys.81.591](https://doi.org/10.1103/RevModPhys.81.591) (2009).
- [4] Jusup, M. *et al.* Social physics. *Physics Reports* **948**, 1–148, DOI: [10.1016/j.physrep.2021.10.005](https://doi.org/10.1016/j.physrep.2021.10.005) (2022).
- [5] D’Orsogna, M. R. & Perc, M. Statistical physics of crime: A review. *Physics of Life Reviews* **12**, 1–21, DOI: [10.1016/j.plrev.2014.11.001](https://doi.org/10.1016/j.plrev.2014.11.001) (2015).
- [6] Kertész, J. & Wachs, J. Complexity science approach to economic crime. *Nature Reviews Physics* **3**, 70–71, DOI: [10.1038/s42254-020-0238-9](https://doi.org/10.1038/s42254-020-0238-9) (2021).
- [7] Granados, O. M. & Nicolás-Carlock, J. R. (eds.) *Corruption Networks: Concepts and Applications* (Springer, Cham, 2021).
- [8] da Cunha, B. R. *Criminofísica: A Ciência das Interações Criminais* (Buqui, Porto Alegre, 2021).
- [9] Newman, M. *Networks: An Introduction* (Orford University Press, New York, 2010).
- [10] Barabási, A.-L. *Network Science* (Cambridge University Press, Cambridge, 2015).
- [11] Luna-Pla, I. & Nicolás-Carlock, J. R. Corruption and complexity: A scientific framework for the analysis of corruption networks. *Applied Network Science* **5**, 13, DOI: [10.1007/s41109-020-00258-2](https://doi.org/10.1007/s41109-020-00258-2) (2020).
- [12] Wachs, J. & Kertész, J. A network approach to cartel detection in public auction markets. *Scientific Reports* **9**, 10818, DOI: [10.1038/s41598-019-47198-1](https://doi.org/10.1038/s41598-019-47198-1) (2019).
- [13] Wachs, J., Fazekas, M. & Kertész, J. Corruption risk in contracting markets: A network science perspective. *International Journal of Data Science and Analytics* **12**, 45–60, DOI: [10.1007/s41060-019-00204-1](https://doi.org/10.1007/s41060-019-00204-1) (2021).

- [14] Garcia-Bedoya, O., Granados, O. & Burgos, J. C. AI against money laundering networks: The Colombian case. *Journal of Money Laundering Control* **24**, 49–62, DOI: [10.1108/JMLC-04-2020-0033](https://doi.org/10.1108/JMLC-04-2020-0033) (2021).
- [15] Colliri, T. & Zhao, L. Analyzing the bills-voting dynamics and predicting corruption-convictions among Brazilian congressmen through temporal networks. *Scientific Reports* **9**, 16754, DOI: [10.1038/s41598-019-53252-9](https://doi.org/10.1038/s41598-019-53252-9) (2019).
- [16] da Cunha, B. R., MacCarron, P., Passold, J. F., dos Santos, L. W., Oliveira, K. A. & Gleeson, J. P. Assessing police topological efficiency in a major sting operation on the dark web. *Scientific Reports* **10**, 73, DOI: [10.1038/s41598-019-56704-4](https://doi.org/10.1038/s41598-019-56704-4) (2020).
- [17] Nicolás-Carlock, J. R. & Luna-Pla, I. Conspiracy of corporate networks in corruption scandals. *Frontiers in Physics* **9**, 301, DOI: [10.3389/fphy.2021.667471](https://doi.org/10.3389/fphy.2021.667471) (2021).
- [18] Calderoni, F., Brunetto, D. & Piccardi, C. Communities in criminal networks: A case study. *Social Networks* **48**, 116–125, DOI: [10.1016/j.socnet.2016.08.003](https://doi.org/10.1016/j.socnet.2016.08.003) (2017).
- [19] Ribeiro, H. V., Alves, L. G. A., Martins, A. F., Lenzi, E. K. & Perc, M. The dynamical structure of political corruption networks. *Journal of Complex Networks* **6**, 989–1003, DOI: [10.1093/comnet/cny002](https://doi.org/10.1093/comnet/cny002) (2018).
- [20] Joseph, J. & Smith, C. M. The ties that bribe: Corruption’s embeddedness in Chicago organized crime. *Criminology* **59**, 671–703, DOI: [10.1111/1745-9125.12287](https://doi.org/10.1111/1745-9125.12287) (2021).
- [21] Solimine, P. C. Political corruption and the congestion of controllability in social networks. *Applied Network Science* **5**, 23, DOI: [10.1007/s41109-020-00263-5](https://doi.org/10.1007/s41109-020-00263-5) (2020).
- [22] Duijn, P. A., Kashirin, V. & Sloot, P. M. The relative ineffectiveness of criminal network disruption. *Scientific Reports* **4**, 4238, DOI: [10.1038/srep04238](https://doi.org/10.1038/srep04238) (2014).
- [23] da Cunha, B. R. & Gonçalves, S. Topology, robustness, and structural controllability of the Brazilian Federal Police criminal intelligence network. *Applied Network Science* **3**, 36, DOI: [10.1007/s41109-018-0092-1](https://doi.org/10.1007/s41109-018-0092-1) (2018).
- [24] Wei, J., Chu, X., Sun, X.-Y., Xu, K., Deng, H.-X., Chen, J., Wei, Z. & Lei, M. Machine learning in materials science. *InfoMat* **1**, 338–358, DOI: [10.1002/inf2.12028](https://doi.org/10.1002/inf2.12028) (2019).
- [25] Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555, DOI: [10.1038/s41586-018-0337-2](https://doi.org/10.1038/s41586-018-0337-2) (2018).

- [26] Artrith, N., Butler, K. T., Coudert, F.-X., Han, S., Isayev, O., Jain, A. & Walsh, A. Best practices in machine learning for chemistry. *Nature Chemistry* **13**, 505–508, DOI: [10.1038/s41557-021-00716-z](https://doi.org/10.1038/s41557-021-00716-z) (2021).
- [27] Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L. & Zdeborová, L. Machine learning and the physical sciences. *Reviews of Modern Physics* **91**, 045002, DOI: [10.1103/RevModPhys.91.045002](https://doi.org/10.1103/RevModPhys.91.045002) (2019).
- [28] Tarca, A. L., Carey, V. J., Chen, X.-w., Romero, R. & Drăghici, S. Machine learning and its applications to biology. *PLOS Computational Biology* **3**, e116, DOI: [10.1371/journal.pcbi.0030116](https://doi.org/10.1371/journal.pcbi.0030116) (2007).
- [29] Molina, M. & Garip, F. Machine learning for sociology. *Annual Review of Sociology* **45**, 27–45, DOI: [10.1146/annurev-soc-073117-041106](https://doi.org/10.1146/annurev-soc-073117-041106) (2019).
- [30] Lim, M., Abdullah, A., Jhanjhi, N. & Khan, M. K. Situation-aware deep reinforcement learning link prediction model for evolving criminal networks. *IEEE Access* **8**, 16550–16559, DOI: [10.1109/ACCESS.2019.2961805](https://doi.org/10.1109/ACCESS.2019.2961805) (2019).
- [31] Calderoni, F., Catanese, S., De Meo, P., Ficara, A. & Fiumara, G. Robust link prediction in criminal networks: A case study of the Sicilian Mafia. *Expert Systems with Applications* **161**, 113666, DOI: [10.1016/j.eswa.2020.113666](https://doi.org/10.1016/j.eswa.2020.113666) (2020).
- [32] Qiao, L.-C., Wang, J.-L., Gao, B.-H., Yang, X.-G., Feng, W.-T., Zhang, Y.-X., Liu, W. & Xu, R.-Y. Utilizing link prediction approach to predict express-related counterfeit cigarette crime cases. In *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, 328–332, DOI: [10.1109/ICCT52962.2021.9657960](https://doi.org/10.1109/ICCT52962.2021.9657960) (IEEE, 2021).
- [33] Goodfellow, I. *Deep Learning* (The Mit Press, Cambridge, Massachusetts, 2016).
- [34] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. & Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **4**, e00938, DOI: [10.1016/j.heliyon.2018.e00938](https://doi.org/10.1016/j.heliyon.2018.e00938) (2018).
- [35] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C. & Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **1**, 57–81, DOI: [10.1016/j.aiopen.2021.01.001](https://doi.org/10.1016/j.aiopen.2021.01.001) (2020).
- [36] Perozzi, B., Al-Rfou, R. & Skiena, S. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’14, 701–710, DOI: [10.1145/2623330.2623732](https://doi.org/10.1145/2623330.2623732) (Association for Computing Machinery, New York, NY, USA, 2014).

- [37] Grover, A. & Leskovec, J. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 855–864, DOI: [10.1145/2939672.2939754](https://doi.org/10.1145/2939672.2939754) (Association for Computing Machinery, New York, NY, USA, 2016).
- [38] Hamilton, W. L. *Graph Representation Learning* (Morgan & Claypool Publishers, San Rafael, California, 2020).
- [39] Cai, H., Zheng, V. W. & Chang, K. C.-C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* **30**, 1616–1637, DOI: [10.1109/TKDE.2018.2807452](https://doi.org/10.1109/TKDE.2018.2807452) (2018). IEEE Transactions on Knowledge and Data Engineering.
- [40] Zhang, D., Yin, J., Zhu, X. & Zhang, C. Network representation learning: A survey. *IEEE Transactions on Big Data* **6**, 3–28, DOI: [10.1109/TBDA.2018.2850013](https://doi.org/10.1109/TBDA.2018.2850013) (2020).
- [41] Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C. & Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy. *arXiv:2005.03675 [cs, stat]* DOI: [10.48550/arXiv.2005.03675](https://doi.org/10.48550/arXiv.2005.03675) (2021).
- [42] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* **32**, 4–24, DOI: [10.1109/TNNLS.2020.2978386](https://doi.org/10.1109/TNNLS.2020.2978386) (2020).
- [43] Martins, A. F., da Cunha, B. R., Hanley, Q. S., Gonçalves, S., Perc, M. & Ribeiro, H. V. Universality of political corruption networks. *Scientific Reports* **12**, 6858, DOI: [10.1038/s41598-022-10909-2](https://doi.org/10.1038/s41598-022-10909-2) (2022).
- [44] Lopes, D. D., Cunha, B. R. d., Martins, A. F., Gonçalves, S., Lenzi, E. K., Hanley, Q. S., Perc, M. & Ribeiro, H. V. Machine learning partners in criminal networks. *Scientific Reports* **12**, 15746, DOI: [10.1038/s41598-022-20025-w](https://doi.org/10.1038/s41598-022-20025-w) (2022).
- [45] Ribeiro, H. V., Lopes, D. D., Pessa, A. A. B., Martins, A. F., da Cunha, B. R., Gonçalves, S., Lenzi, E. K., Hanley, Q. S. & Perc, M. Deep learning criminal networks. *Chaos, Solitons & Fractals* **172**, 113579, DOI: [10.1016/j.chaos.2023.113579](https://doi.org/10.1016/j.chaos.2023.113579) (2023).
- [46] Todos los Casos de Corrupción en España. <https://www.Casos-Aislados.com/index.php> (Último acesso em: 18/10/2020).
- [47] Barabási, A.-L. & Pósfai, M. *Network Science* (Cambridge University Press, Cambridge, United Kingdom, 2016).
- [48] Newman, M. *Networks: An Introduction* (Oxford University Press, 2010).

- [49] Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America* **99**, 7821–7826, DOI: [10.1073/pnas.122653799](https://doi.org/10.1073/pnas.122653799) (2002).
- [50] Newman, M. E. J. Detecting community structure in networks. *The European Physical Journal B* **38**, 321–330, DOI: [10.1140/epjb/e2004-00124-y](https://doi.org/10.1140/epjb/e2004-00124-y) (2004).
- [51] Newman, M. E. J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* **103**, 8577–8582, DOI: [10.1073/pnas.0601602103](https://doi.org/10.1073/pnas.0601602103) (2006).
- [52] Calderoni, F., Brunetto, D. & Piccardi, C. Communities in criminal networks: A case study. *Social Networks* **48**, 116–125, DOI: [10.1016/j.socnet.2016.08.003](https://doi.org/10.1016/j.socnet.2016.08.003) (2017).
- [53] da Cunha, B. R., González-Avella, J. C. & Gonçalves, S. Fast fragmentation of networks using module-based attacks. *PLOS ONE* **10**, e0142824, DOI: [10.1371/journal.pone.0142824](https://doi.org/10.1371/journal.pone.0142824) (2015).
- [54] Peixoto, T. P. Bayesian stochastic blockmodeling. *arXiv [cond-mat, physics:physics, stat]* 289–332, DOI: [10.1002/9781119483298.ch11](https://doi.org/10.1002/9781119483298.ch11) (2019).
- [55] Bunde, A. & Havlin, S. (eds.) *Fractals and Disordered Systems* (Springer-Verlag, Berlin, 1996).
- [56] Menardi, G. & Torelli, N. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery* **28**, 92–122, DOI: [10.1007/s10618-012-0295-5](https://doi.org/10.1007/s10618-012-0295-5) (2014).
- [57] Hamilton, W., Ying, Z. & Leskovec, J. Inductive representation learning on large graphs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. & Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc., 2017).
- [58] Maeda, E. E., Haapasaari, P., Helle, I., Lehikoinen, A., Voinov, A. & Kuikka, S. Black boxes and the role of modeling in environmental policy making. *Frontiers in Environmental Science* 63, DOI: [10.3389/fenvs.2021.629336](https://doi.org/10.3389/fenvs.2021.629336) (2021).
- [59] Possati, L. M. Algorithmic unconscious: why psychoanalysis helps in understanding AI. *Palgrave Communications* **6**, 1–13, DOI: [10.1057/s41599-020-0445-0](https://doi.org/10.1057/s41599-020-0445-0) (2020).
- [60] Le Merrer, E. & Trédan, G. Remote explainability faces the bouncer problem. *Nature Machine Intelligence* **2**, 529–539, DOI: [10.1038/s42256-020-0216-z](https://doi.org/10.1038/s42256-020-0216-z) (2020).

- [61] Molnar, C., Casalicchio, G. & Bischl, B. Interpretable machine learning – A brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 417–431, DOI: [10.1007/978-3-030-65965-3_28](https://doi.org/10.1007/978-3-030-65965-3_28) (2020).
- [62] Li, Y., Zhou, J., Verma, S. & Chen, F. A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint arXiv:2207.12599* DOI: [10.48550/arXiv.2207.12599](https://doi.org/10.48550/arXiv.2207.12599) (2022).
- [63] Kang, H. & Park, H. Providing node-level local explanation for node2vec through reinforcement learning. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining* (Association for Computing Machinery, New York, NY, USA, 2022).
- [64] Newman, M. E. J. Mixing patterns in networks. *Physical Review E* **67**, 026126, DOI: [10.1103/PhysRevE.67.026126](https://doi.org/10.1103/PhysRevE.67.026126) (2003).
- [65] Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* **105**, 1118–1123, DOI: [10.1073/pnas.0706851105](https://doi.org/10.1073/pnas.0706851105) (2008).
- [66] Rosvall, M., Axelsson, D. & Bergstrom, C. T. The map equation. *The European Physical Journal Special Topics* **178**, 13–23, DOI: [10.1140/epjst/e2010-01179-1](https://doi.org/10.1140/epjst/e2010-01179-1) (2009).
- [67] Huffman, D. A. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* **40**, 1098–1101, DOI: [10.1109/JRPROC.1952.273898](https://doi.org/10.1109/JRPROC.1952.273898) (1952).
- [68] Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal* **27**, 379–423, DOI: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x) (1948).
- [69] Infomap: Infomap network clustering algorithm. <https://mapequation.github.io/infomap/> (Último acesso em: 03/01/2023).
- [70] Efron, B. & Tibshirani, R. J. *An Introduction to the Bootstrap* (Chapman and Hall/CRC, London, 1994).
- [71] James, G., Witten, D., Hastie, T. & Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R* (Springer, New York, 2014).
- [72] Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition* (Springer, New York, NY, 2009).
- [73] Bishop, C. M. *Pattern Recognition and Machine Learning* (Springer, New York, 2011).

- [74] Geron, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O'Reilly Media, Beijing China; Sebastopol, CA, 2019).
- [75] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).
- [76] Müller, A. & Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists* (O'Reilly Media, Sebastopol, CA, 2016).
- [77] Hamilton, W. L., Ying, R. & Leskovec, J. Representation Learning on Graphs: Methods and Applications (2018). ArXiv:1709.05584 [cs].
- [78] Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient Estimation of Word Representations in Vector Space (2013). ArXiv:1301.3781 [cs].
- [79] Cohen, E. node2vec: Implementation of the node2vec algorithm. <https://github.com/eliorc/node2vec>. (Último acesso em: 04/04/2023).
- [80] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. & Mei, Q. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077, DOI: [10.1145/2736277.2741093](https://doi.org/10.1145/2736277.2741093) (2015).
- [81] Kullback, S. & Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics* **22**, 79–86 (1951).
- [82] García-Pérez, G., Allard, A., Serrano, M. A. & Boguna, M. Mercator: Uncovering faithful hyperbolic embeddings of complex networks. *New Journal of Physics* **21**, 123033, DOI: [10.1088/1367-2630/ab57d2](https://doi.org/10.1088/1367-2630/ab57d2) (2019).
- [83] Penrose, M. *Random Geometric Graphs*. Oxford Studies in Probability (Oxford University Press, Oxford, New York, 2003).
- [84] McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* DOI: [10.48550/arXiv.1802.03426](https://doi.org/10.48550/arXiv.1802.03426) (2018).
- [85] Altman, N. & Krzywinski, M. The curse(s) of dimensionality. *Nature Methods* **15**, 399–400, DOI: [10.1038/s41592-018-0019-x](https://doi.org/10.1038/s41592-018-0019-x) (2018).
- [86] McInnes, L., Healy, J., Saul, N. & Grossberger, L. umap-learn: Uniform Manifold Approximation and Projection.
- [87] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L. & Leskovec, J. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data*

Mining, KDD '18, 974–983, DOI: [10.1145/3219819.3219890](https://doi.org/10.1145/3219819.3219890) (Association for Computing Machinery, New York, NY, USA, 2018).

- [88] Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, 8024–8035 (Curran Associates, Inc., 2019).
- [89] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv* DOI: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980) (2014).