Caracterização de classes e detecção de *outliers* em redes complexas

Lilian Berton

Caracterização de classes e detecção de *outliers* em redes complexas

Lilian Berton

Orientador: **Prof. Dr. Zhao Liang**

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

USP – São Carlos
Junho de 2011

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

Berton, Lilian
B547c Caracterização de classes e detecção de outliers em redes complexas / Lilian Berton; orientador Zhao Liang -- São Carlos, 2011.
89 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2011.

1. Detecção de outlier. 2. Redes complexas. 3. Caracterização de classes. I. Liang, Zhao, orient. II. Título.

Agradecimentos

A minha família pelo amor incondicional a mim concedido, pelas bases que me deram, pela educação, pelas oportunidades que me proporcionaram e mesmo estando distantes estão sempre presentes em meu coração.

Ao professor Zhao Liang pela orientação durante o mestrado, pela paciência quando eu não sabia fazer o certo e pelos conselhos para que eu melhorasse.

Aos professores Maria José Castanho, Marlon Soares e Fábio Hernandes da Universidade Estadual do Centro-Oeste, pela orientação nas iniciações científicas, pelos conselhos e pela amizade. Vocês que me inspiraram e me incentivaram para que eu fizesse o mestrado.

Aos professores Alneu Lopes, Mauricio Figueiredo e Ivan Nunes pelas observações e sugestões concedidas no exame de qualificação e na defesa.

Ao Didier Vega por todo apoio concedido durante o período do mestrado, por sua imensa ajuda, pela paciência, seu carinho e seu amor.

Ao Marcos Aurélio Pedroso pela preciosa amizade que tem me proporcionado ao longo dos anos que o tenho conhecido e que mesmo longe continua presente.

Aos colegas João Bertini, Robson Motta, Thiago Christiano, Cássio Martins e Andrés Coca que me auxiliaram esclarecendo dúvidas, sugerindo ferramentas e dando sugestões.

Aos colegas Bilzã Araújo e Jean Huertas pelo trabalho desenvolvido em conjunto e pelo companheirismo nas disciplinas que realizamos juntos.

A todos os colegas do Biocomp que tem me ajudado emprestando o computador para realizar experimentos ou esclarecendo dúvidas. Em especial agradeço ao Paulo Gabriel pelas correções realizadas nesta dissertação, por toda ajuda concedida e por sua amizade e ao Thiago Cupertino pelas sugestões dadas e por sua amizade.

As amigas Ana Claudia Ortega, Debora Medeiros e Glenda Botelho pelo companheirismo seja nas horas de estudo ou de distração e pelas experiências compartilhadas.

Ao programa de pós-graduação do ICMC-USP.

À FAPESP pelo suporte financeiro durante boa parte desse mestrado.

Agradeço especialmente a Deus por ter me proporcionado chegar até aqui e por ter colocado todas estas pessoas no meu caminho. Por ter me protegido e me amparado sempre. Obrigada por tudo!

Resumo

As redes complexas surgiram como uma nova e importante maneira de representação e abstração de dados capaz de capturar as relações espaciais, topológicas, funcionais, entre outras características presentes em muitas bases de dados. Dentre as várias abordagens para a análise de dados, destacam-se a classificação e a detecção de *outliers*. A classificação de dados permite atribuir uma classe aos dados, baseada nas características de seus atributos e a detecção de *outliers* busca por dados cujas características se diferem dos demais. Métodos de classificação de dados e de detecção de *outliers* baseados em redes complexas ainda são pouco estudados. Tendo em vista os benefícios proporcionados pelo uso de redes complexas na representação de dados, o presente trabalho apresenta o desenvolvimento de um método baseado em redes complexas para detecção de *outliers* que utiliza a caminhada aleatória e um índice de dissimilaridade. Este método possibilita a identificação de diferentes tipos de *outliers* usando a mesma medida. Dependendo da estrutura da rede, os vértices *outliers* podem ser tanto aqueles distantes do centro como os centrais, podem ser *hubs* ou vértices com poucas ligações. De um modo geral, a medida proposta é uma boa estimadora de vértices *outliers* em uma rede, identificando, de maneira adequada, vértices com uma estrutura diferenciada ou com uma função especial na rede. Foi proposta também uma técnica de construção de redes capaz de representar relações de similaridade entre classes de dados, baseada em uma função de energia que considera medidas de pureza e extensão da rede. Esta rede construída foi utilizada para caracterizar mistura entre classes de dados. A caracterização de classes é uma questão importante na classificação de dados, porém ainda é pouco explorada. Considera-se que o trabalho desenvolvido é uma das primeiras tentativas nesta direção.

Abstract

Complex networks have emerged as a new and important way of representation and data abstraction capable of capturing the spatial relationships, topological, functional, and other features present in many databases. Among the various approaches to data analysis, we highlight classification and outlier detection. Data classification allows to assign a class to the data based on characteristics of their attributes and outlier detection search for data whose characteristics differ from the others. Methods of data classification and outlier detection based on complex networks are still little studied. Given the benefits provided by the use of complex networks in data representation, this study developed a method based on complex networks to detect outliers based on random walk and on a dissimilarity index. The method allows the identification of different types of outliers using the same measure. Depending on the structure of the network, the vertices outliers can be either those distant from the center as the central, can be hubs or vertices with few connections. In general, the proposed measure is a good estimator of outlier vertices in a network, properly identifying vertices with a different structure or a special function in the network. We also propose a technique for building networks capable of representing similarity relationships between classes of data based on an energy function that considers measures of purity and extension of the network. This network was used to characterize mixing among data classes. Characterization of classes is an important issue in data classification, but it is little explored. We consider that this work is one of the first attempts in this direction.

Sumário

Agradecimentos ..... **i**

Resumo ..... **iii**

Abstract ..... **v**

Sumário ..... **vii**

Lista de Figuras ..... **ix**

Lista de Tabelas ..... **xi**

1 Introdução **1**

1.1 Motivação 2

1.2 Objetivos 4

1.3 Metodologia 4

1.4 Organização do Documento 5

2 Revisão sobre redes complexas **6**

2.1 Algumas medidas de redes complexas 7

2.2 Tipos de redes 12

2.2.1 Redes aleatórias 12

2.2.2 Redes mundo pequeno 12

2.2.3 Redes livre de escala 13

2.3 Métodos de detecção de comunidades em redes complexas 15

2.3.1 *Betweenness* 15

2.3.2 Modularidade 16

2.3.3 Caminhada aleatória 17

2.3.4 Competição de partículas 19

2.4 Otimização em redes complexas 21

3 Revisão sobre detecção de *outliers* e classificação de dados 25
3.1 Aprendizado de máquina 25
3.2 Detecção de *outliers* 26
3.3 Classificação de dados 30
3.3.1 Árvores de decisão 31
3.3.2 Redes neurais artificiais 32
3.3.2.1 Perceptron 32
3.3.2.2 Redes neurais artificiais multicamadas 33
3.3.3 *K*-vizinhos mais próximos 34
3.3.4 Classificador Naïve Bayes 35
3.3.5 Classificação relacional 36
3.3.5.1 Inferência coletiva 37
3.3.5.2 Classificadores relacionais 39
3.3.6 Classificação baseada em rede *K*-associados 41
3.3.6.1 Rede *K*-associados e medida de pureza 41
3.3.6.2 Classificador *K*-associados 44

4 Detecção de *outliers* em redes complexas 46
4.1 Medida de distância e índice de dissimilaridade 46
4.2 O método de detecção de *outliers* 48
4.3 Detecção de *outliers* em redes artificiais 48
4.4 Detecção de *outliers* em redes reais 51
4.5 Discussão do método 54

5 Caracterização de classes via otimização em redes complexas 55
5.1 O método proposto 55
5.2 Simulações em redes artificiais 59
5.3 Simulações em redes reais 71
5.4 Discussão do método 75

6 Conclusões 77
6.1 Contribuições 78
6.2 Trabalhos futuros 79

Referências Bibliográficas 80

Lista de Figuras

Figura 1: O modelo de Watts-Strogatz para redes mundo pequeno..... 13
Figura 2: Rede de colaboração entre cientistas 14
Figura 3: Distribuição do grau para várias redes reais 14
Figura 4: Dendograma das comunidades encontradas pelo algoritmo de modularidade Q. 17
Figura 5: (A) Densidade de *links*, (B) energia, (C) coef. de clust. e (D) distância. 23
Figura 6: Redes ótimas para determinados valores de λ.. 24
Figura 7: Exemplo simples de *outliers* em um conjunto de dados 2-D. 28
Figura 8: Uma árvore de decisão para um problema de classificação de mamíferos 31
Figura 9: Neurônio perceptron 33
Figura 10: Rede neural com duas camadas 34
Figura 11: Exemplo de 1, 2 e 3-vizinhos mais próximos do dado *x* 35
Figura 12: Redes *K*-associados.. 42
Figura 13: Um exemplo de um componente "puro" com 5 vértices e *K*= 3.. 43
Figura 14: Método de detecção de *outlier* aplicado em uma cadeia com 5 vértices..... 49
Figura 15: Método de detecção de *outlier* aplicado em uma cadeia com 9 vértices..... 49
Figura 16: Método de detecção de *outlier* aplicado em uma cadeia com 13 vértices..... 50
Figura 17: Método de detecção de *outlier* aplicado à rede clube de karate 52
Figura 18: Método de detecção de *outlier* aplicado à sub-rede de colaboração científica..... 53
Figura 19: Base de dados *Gaussianas 1-2-3* 59
Figura 20: Redes formadas para a base *Gaussianas 1-2-3* 60
Figura 21: Pureza para as redes formadas a partir das bases *Gaussianas 1-2-3* 61
Figura 22: Extensão para as redes formadas a partir das bases *Gaussianas 1-2-3*. 61
Figura 23: Energia para as redes formadas a partir das bases *Gaussianas 1-2-3*..... 62
Figura 24: Base de dados *Bananas 1-2-3*..... 63
Figura 25: Redes formadas para a base *Bananas 1-2-3* 63
Figura 26: Pureza para as redes formadas a partir das bases *Bananas 1-2-3*..... 64
Figura 27: Extensão para as redes formadas a partir das bases *Bananas 1-2-3*..... 64
Figura 28: Energia para as redes formadas a partir das bases *Bananas 1-2-3* 65
Figura 29: Base de dados *Dispersão 1-2-3* 65

Figura 30: Redes formadas para a base *Dispersão 1-2-3*..... 66
Figura 31: Pureza para as redes formadas a partir das bases *Dispersão 1-2-3* 67
Figura 32: Extensão para as redes formadas a partir das bases *Dispersão 1-2-3*. 67
Figura 33: Energia para as redes formadas a partir das bases *Dispersão 1-2-3*..... 68
Figura 34: Base de dados *8-Gaussianas* e *Multiclasse* 68
Figura 35: Redes formadas para a base *8-Gaussianas* 69
Figura 36: Redes formadas para a base *Multiclasse* 69
Figura 37: Pureza para as redes das bases *8-Gaussianas* e *Dispersão*..... 70
Figura 38: Extensão para as redes das bases *8-Gaussianas* e *Dispersão*..... 70
Figura 39: Energia para as redes das bases *8-Gaussianas e Dispersão*..... 71
Figura 40: Base de dados Iris..... 72
Figura 41: Redes formadas para a base Iris 72
Figura 42: Base de dados Glass 72
Figura 43: Redes formadas para a base Glass..... 73
Figura 44: Base de dados Zoo..... 73
Figura 45: Redes formadas para a base Zoo 73
Figura 46: Pureza para as redes das bases Iris, Glass e Zoo 74
Figura 47: Extensão para as redes das bases Iris, Glass e Zoo 74
Figura 48: Energia para as redes das bases Iris, Glass e Zoo..... 75

Lista de Tabelas

Tabela 1: Matriz de confusão para um problema de 2 classes 30
Tabela 2: *Rank* dos 10 vértices com maior *score* de *outlier* na rede clube de karate 52
Tabela 3: *Rank* dos 10 vértices com maior *score* de *outlier* na rede colaboração científica ... 53

Capítulo 1

Introdução

O avanço tecnológico tem permitido o armazenamento de grandes quantidades de dados, entretanto a exploração de uma informação relevante ou específica tem apresentado muitas dificuldades (Witten e Frank, 2000; Tan, Steinbach e Kumar, 2006). Algumas dificuldades encontradas por técnicas de análise tradicionais em novos conjuntos de dados são apresentadas a seguir (Tan, Steinbach e Kumar, 2006).

- Escalabilidade: conjuntos de dados com tamanhos em terabytes e até petabytes estão se tornando comuns, exigindo que os algoritmos sejam escaláveis. Nesse sentido, pode-se tornar necessário o desenvolvimento de algoritmos paralelos ou distribuídos.
- Alta dimensionalidade: dados com centenas ou milhares de atributos também estão sendo comuns, exigindo algoritmos que funcionem bem para alta dimensão.
- Dados heterogêneos e complexos: objetos de dados mais complexos têm emergido, como coleções de páginas web contendo textos semi-estruturados e *hyperlinks*, dados de DNA com estrutura sequencial e tri-dimensional, dados climáticos com várias medidas (como temperatura e pressão) obtidos de vários locais da superfície terrestre exigem que as técnicas desenvolvidas levem em consideração relações nos dados, conectividade do grafo, etc.
- Distribuição e propriedade dos dados: algumas vezes, os dados a serem analisados não estão armazenados em um único local, ou não são de propriedade de uma única organização, requerendo o desenvolvimento de técnicas distribuídas.

Com o objetivo de resolver essas dificuldades, pesquisadores de diferentes disciplinas começaram a desenvolver ferramentas mais eficientes e escaláveis que pudessem suportar diversos tipos de dados, culminando no surgimento da área de mineração de dados.

A mineração de dados (*Data Mining* - DM) é uma parte integrante da descoberta de conhecimento em base de dados (*Knowledge Discovery in Database* - KDD), que corresponde ao processo geral de conversão de dados em informações úteis (Witten e Frank, 2000). Técnicas de mineração de dados são aplicadas em grandes bases de dados para encontrar padrões originais e úteis, que fora isso permaneceriam desconhecidos. Essas técnicas podem prover também a capacidade de predizer o resultado de uma observação futura (Tan, Steinbach e Kumar, 2006).

A mineração de dados está ligada com diversas áreas, uma delas é o aprendizado de máquina (*Machine Learning* - ML) a qual desenvolve algoritmos que melhoram seu desempenho automaticamente com a experiência (Mitchell, 1997).

A maioria dos algoritmos em aprendizado de máquina utiliza uma representação proposicional dos dados (tabela atributo-valor) e esta não pode representar adequadamente domínios envolvendo múltiplas entidades, bem como as relações entre elas (Raedt, 2008). Outra forma que pode ser usada para representação dos dados é a relacional, a qual pode ser mais rica e interessante já que apresenta além das informações dos objetos, a relação existente entre eles (Raedt, 2008).

Objetos com relações entre si constituem uma rede. As redes que modelam sistemas complexos, referidas como redes complexas, são compostas por milhares ou até bilhões de vértices, possuem topologia não trivial e são capazes de capturar as relações espaciais, topológicas, funcionais, entre outras características presentes em muitas bases de dados (Albert e Barabási, 2002; Newman, 2003).

1.1 Motivação

Considerando a utilização de redes complexas como uma ferramenta de representação e manipulação dos dados, muitas vantagens podem ser pertinentes ao uso deste tipo de rede na análise de dados. A principal vantagem é a possibilidade de tratar grandes bases de dados e, desse modo, poder analisá-las por meio de um tratamento estatístico apropriado. Uma rede pode ser estudada sob diversas abordagens, o que possibilita a combinação de estrutura, dinâmica, evolução, entre outras características para melhor representação dos dados e consequentemente melhor análise. Além disso, o uso de redes complexas para representar os dados pode revelar estruturas topológicas e dependências antes não observadas.

Métodos de agrupamento de dados baseados em redes complexas, também conhecidos como detecção de comunidades, têm sido extensivamente explorados na literatura. Tais comunidades podem ser definidas como grupos de vértices da rede densamente conectados, enquanto que conexões entre vértices de grupos diferentes são esparsas (Newman e Girvan, 2004). Muitas técnicas foram desenvolvidas baseadas em diversas ideias para detecção de comunidades. Alguns exemplos podem ser encontrados em: (Zhou, 2003a; Zhou, 2003b; Newman, 2004; Newman e Girvan, 2004; Reichardt e Bornholdt, 2004; Boccaletti et al., 2007; Quiles et al., 2008). De maneira geral, a estrutura em comunidades revela similaridade por meio de conexões entre os vértices pertencentes a um mesmo grupo. Essas similaridades, por sua vez, podem revelar grupos nos dados em problemas de agrupamento e, de maneira análoga, podem evidenciar classes em problemas de classificação. Desse modo, as redes complexas também estão sendo usadas com sucesso na classificação de dados.

A classificação de dados lida com a detecção automática de padrões em conjuntos de dados. Por padrões, entendem-se relações, regularidades ou estruturas inerentes a alguns conjuntos de dados (Mitchell, 1997). Por meio da detecção de padrões significantes nos dados disponíveis (conjunto de treinamento), espera-se que um classificador possa realizar predições de classes em novos dados de entrada. Existem muitos problemas importantes que podem ser resolvidos utilizando-se dessa abordagem, abrangendo diversas áreas como bioinformática (Baldi e Brunak, 1998), mineração de dados (Cook e Holder, 2000), reconhecimento de escrita (LeCun et al., 1989), entre outras.

Buscando ampliar as possibilidades dos classificadores, Lopes et al. (2009) propuseram uma técnica denominada rede *K*-associados, a qual constrói uma representação relacional dos dados via redes, nos quais os objetos são os vértices e as arestas ligam objetos similares que possuem a mesma classe, a partir das redes geradas faz-se a classificação. Esta rede proposta leva em consideração uma medida de pureza, a qual é definida formalmente na Seção 3.3.6. Porém, a pureza não considera a extensão dos componentes de forma explícita e a técnica tende a favorecer a formação de muitos componentes pequenos. Isso motivou o estudo da técnica com mais detalhes a fim de buscar o aperfeiçoamento dessa medida. Por fim, com base nas redes *K*-associados, propomos uma nova técnica para construção de redes e a utilizamos para caracterizar mistura de classes em conjuntos de dados.

Outro problema importante em DM trata da detecção de *outliers*, a qual busca por padrões nos dados que não estão de acordo com o comportamento esperado. A detecção de *outliers* tem sido extensivamente utilizada em diversas aplicações, como detecção de fraudes em cartões de crédito, detecção de intrusão em redes, diagnóstico de falhas, processamento de pedido de empréstimo, entre outros. Muitas técnicas têm sido desenvolvidas para a detecção

de *outliers* e a maioria delas apresentam restrições. Por exemplo: técnicas baseadas em classificação dependem da disponibilidade de um conjunto de dados rotulados para o treinamento; técnicas estatísticas assumem determinado conhecimento sobre as características e distribuição dos dados; técnicas baseadas em distância são normalmente custosas e não podem ser aplicadas em cenários onde a complexidade computacional é uma questão importante (Chandola, Banerjee e Kumar, 2007).

Entretanto pouco tem sido estudado sobre a identificação de *outliers* em redes complexas, o que pode ser importante em diversas situações. Em redes de comunicações, por exemplo, vértices *outliers* podem ser os mais vulneráveis e a sua identificação pode auxiliar na melhoria da segurança (Lai et al., 2005); em redes biológicas, como redes de interação entre proteínas, vértices *outliers* correspondem as proteínas com funções especiais (Palla et al., 2005).

Com base nisto, procurou-se neste trabalho propor uma técnica de detecção de *outliers* baseada em redes complexas que não apresentasse as restrições dos métodos tradicionais.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver uma técnica baseada em redes complexas para detecção de *outliers* e outra para o estudo de classes em conjuntos de dados. Os objetivos específicos são:

- desenvolver uma técnica de identificação de vértices *outliers* em redes, baseada na caminhada aleatória de uma partícula Browniana e em um índice de dissimilaridade apresentado por Zhou (2003b);
- aplicar a medida de identificação de *outliers* em algumas redes e analisar os resultados obtidos;
- desenvolver uma técnica de construção de redes considerando-se medidas de pureza e extensão da rede;
- aplicar a técnica de construção de redes em algumas bases de dados para caracterizar mistura entre classes de dados.

1.3 Metodologia

Para o desenvolvimento da medida de identificação de *outliers* considerou-se o movimento de uma partícula Browniana em uma rede, o qual pode ser interpretado como uma caminhada aleatória. A medida de distância obtida a partir da caminhada aleatória é usada por Zhou (2003a) para detecção de comunidades em redes, este trabalho é apresentado na Seção

2.3.3. Aqui, utilizou-se a medida de distância e o índice de dissimilaridade para identificar vértices *outliers* na rede.

Aplicou-se o método em algumas redes artificiais e na rede clube de karate, registrada por Zachary (1977) e em uma sub-rede de colaboração científica, compilada por Newman (2006).

Para o desenvolvimento da técnica de construção de redes, utilizou-se uma função de energia que reúne *pureza* e *extensão* da rede. A rede é construída a partir de um conjunto de dados proposicionais e baseada nas relações de similaridade entre os vértices da mesma classe, de modo que cada vértice irá alterar suas conexões com um *k*-vizinho mais próximo, se esta alteração maximizar a *função de energia*. O processo de maximização de uma função de energia foi inspirado pelo trabalho de Cancho e Solé (2003), o qual é apresentado na Seção 2.4.

A técnica foi aplicada em alguns conjuntos de dados artificiais e reais para analisar seu comportamento na caracterização das classes dos dados. Procurou-se analisar a mistura dos dados, para isso, utilizaram-se dados com diferentes formatos e quantidade de classes.

1.4 Organização do documento

No que segue, no Capítulo 2, são apresentados alguns tópicos relevantes sobre redes complexas, mais especificamente são descritos os principais tipos de redes complexas, algumas medidas, alguns métodos de detecção de comunidades baseados em redes complexas e um método de otimização. No Capítulo 3, conceitos sobre aprendizado de máquina são apresentados, com foco na detecção de *outliers* e na classificação de dados. São apresentadas as principais técnicas de classificação e as propriedades da rede *K*-associados e do classificador baseado nela. No Capítulo 4, são apresentadas a medida proposta para detecção de *outliers* em redes complexas e os resultados obtidos com a aplicação do método em algumas redes artificiais e reais. No Capítulo 5, são descritas a proposta de construção de redes para caracterização de mistura de classes em conjuntos de dados e os resultados obtidos com a aplicação em algumas redes artificiais e reais. Por fim, no Capítulo 6, são apresentadas as conclusões e propostas de trabalhos futuros.

Capítulo 2

Revisão sobre redes complexas

A partir da metade da década de 90, as redes de Internet e a *World Wide Web* (WWW) começaram a destacar-se. Essas redes eram grandes, dinâmicas e possuíam complexas arquiteturas, surgindo assim a necessidade de entendimento de seu funcionamento e organização global (Dorogovtsev e Mendes, 2003).

Com a disponibilidade de computadores que permitiam analisar dados em uma escala muito maior do que era possível no passado, alguns autores começaram a estudá-las e observaram que muitas dessas redes apresentavam distribuição de grau pela lei da potência, destacam-se os estudos de Albert, Jeong e Barabási (1999) e Huberman e Adamic (1999) na WWW, e Faloutsos, Faloutsos e Faloutsos (1999), Govindan e Tangmunarunkit (2000) na Internet.

Outros estudos mostraram que diversas redes naturais e artificiais também têm distribuição de grau dada pela lei da potência. Por exemplo, redes de citações científicas (Redner, 1998), redes de reações metabólicas (Jeong et al., 2000) redes de interação de proteínas (Jeong et al., 2001), entre outras.

Essas redes, denominadas redes complexas, apresentam propriedades topológicas interessantes que não são encontradas em grafos mais simples. Em Chung e Lu (2006) são destacadas como principais características desse tipo de rede:

- volume: o tamanho das redes geralmente abrange centenas ou milhares de vértices. Abordagens por força brutas não são viáveis, portanto deve-se usar um número relativamente pequeno de parâmetros para capturar as características da rede;
- esparsa: o número de arestas é dentro de um múltiplo pequeno do número de vértices.

- Fenômeno mundo pequeno: é usado para se referir a duas propriedades: distância pequena (dois estranhos geralmente são conectados por um número pequeno de conhecidos mútuos) e efeito *clustering* (duas pessoas que compartilham um vizinho em comum são mais propensas a se conhecerem);
- distribuição do grau pela lei da potência: o grau de um vértice é o número de vértices adjacente a ele. A lei da potência garante que o número de vértices com grau k é proporcional a $k^{-\gamma}$ para algum expoente $\gamma \geq 1$.

A seguir são apresentadas algumas noções básicas de redes complexas. Na Seção 2.1 são apresentadas algumas medidas relativas às redes complexas, na Seção 2.2 são mostrados os principais modelos de redes, a Seção 2.3 traz alguns métodos para detecção de comunidades em redes e na Seção 2.4 é exposto um modelo de otimização em redes.

2.1 Algumas medidas de redes complexas

Uma rede pode ser representada por um grafo, onde cada elemento da rede é mapeado para um vértice (nó), e a interação entre dois vértices é representada por uma aresta (arco), cujo peso é relacionado com a força da interação. Esse grafo pode ser denotado formalmente por $G = (V, E, W)$, sendo $V = \{v_1, v_2, ..., v_N\}$ o conjunto de N vértices, $E = \{e_1, e_2, ..., e_M\}$ o conjunto de M arestas e $W = \{w_1, w_2, ..., w_M\}$ os pesos associados a cada aresta, no caso de redes ponderadas.

Um grafo pode ser representado por uma matriz de adjacência A_{NXN}, cuja entrada a_{ij} ($i, j = 1, ..., N$) é igual a 1 quando a aresta e_{ij} existir, e zero caso contrário.

A seguir são apresentadas algumas medidas de redes complexas:

Grau: o grau de um vértice representa o número de ligações que o vértice possui e é definido em termos de uma matriz de adjacência conforme a Equação (1).

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

Grau médio: o grau médio $\langle g \rangle$ representa a média do grau de todos os vértices da rede.

$$\langle g \rangle = \frac{1}{N} \sum_{i=1}^{N} g_i \tag{2}$$

Distribuição de grau: um grafo pode ser caracterizado por sua distribuição de grau, P_k, definida como a fração de vértices no grafo com grau k. Pode-se fazer um gráfico de P_k para se obter informações de como o grau está distribuído entre os vértices. Através do grau dos

vértices é possível encontrar vértices altamente conectados, denominados *hubs*, os quais podem ter uma função importante em uma rede.

Componente: o componente ao qual um vértice *i* pertence é um conjunto de vértices que podem ser alcançados a partir de *i*, por meio de percursos entre as arestas do grafo.

Caminho: os percursos sobre os vértices de uma rede formam sequências de arestas em que todas são distintas. Se uma sequência de arestas, além de ser distinta, não repetir os vértices esta cadeia é denominada caminho.

Menor caminho: partindo-se de um vértice *i*, é possível determinar quantos passos são necessários para se chegar a qualquer vértice *j* de uma rede. Quando se encontra o menor número de passos necessários, este caminho percorrido é chamado de menor caminho entre *i* e *j*, ou *caminho geodésico* se a rede for não ponderada.

Diâmetro: o menor caminho também tem um importante papel na caracterização de estruturas internas de um grafo. Pode-se representar o conjunto de menores caminhos de um grafo como uma matriz *D*, na qual d_{ij} é o caminho geodésico do vértice *i* até o vértice *j*. O maior valor de d_{ij} é chamado de *diâmetro* da rede. Uma medida típica de separação entre dois vértices na rede é dada pela média dos caminhos geodésicos de todos os pares de vértices:

$$L = \frac{1}{N(N-1)} \sum_{i,j \in N, i \neq j} d_{ij} \tag{3}$$

Closeness: um vértice pode ter alto grau, mas pertencer a um grupo isolado, não sendo bem conectado numa escala global. Uma medida que explora a relação de um vértice com todos os vértices da rede é chamada grau de proximidade (*closeness*) (Wasserman e Faust, 1994). O grau de proximidade c_i de um vértice v_i é o inverso da média dos caminhos geodésicos para todos os outros vértices da rede, conforme a Equação (4).

$$c_i = \frac{N}{\sum_{j=1}^{N} d_{ij}} \tag{4}$$

na qual *N* é o número de vértices que possuem um caminho para o vértice v_i não contendo o próprio vértice. Quanto maior o valor de c_i, menor a distância do vértice v_i para os outros vértices da rede.

Betweenness: a comunicação entre dois vértices não adjacentes, *j* e *k*, depende dos vértices pertencentes ao caminho que conecta *j* a *k*. Uma medida da relevância de um dado vértice pode ser obtida contando-se o número de caminhos geodésicos que passam por ele. Esta medida é conhecida por *betweenness* do vértice (Freeman, 1977). O *betweenness* b_i de um vértice *i* é definido como:

$$b_i = \sum_{j,k \in N, j \neq k} \frac{n_{jk}(i)}{n_{jk}} \tag{5}$$

na qual n_{jk} é o número de menores caminhos conectando *j* e *k*, e $n_{jk}(i)$ é o número de menores caminhos conectando *j* e *k* que passam por *i*.

Betweenness indica se um vértice é importante ou não para o trafego na rede. O conceito de *betweenness* pode ser estendido também para as arestas. O *betweenness* de uma aresta é definido como o número de menores caminhos entre pares de vértices que passam por uma dada aresta. Pode ser usado para detectar estruturas hierárquicas em redes, ver Seção 2.3.1.

Centralidade por caminhada aleatória (*random walk centrality*): o menor caminho apenas pode ser encontrado por meio do conhecimento da conectividade global de cada vértice, o que muitas vezes é impraticável. Em casos nos quais apenas a conectividade local de um vértice é conhecida, pode-se usar a caminhada aleatória para estudar a estrutura da rede. Uma medida que quantifica quão centralizado um vértice *i* está, levando-se em conta seu potencial para receber informações randomicamente difundidas na rede, é chamada centralidade por caminhada aleatória, proposta por Noh e Rieger (2004).

Considerando-se uma rede finita que consiste de vértices $i = 1, \cdots, N$ e arestas conectando-os. Assume-se que a rede é conexa (isto é, existe um caminho entre cada par de vértices (*i*, *j*)), caso contrário considera-se cada componente separadamente. A conectividade é representada pela matriz de adjacência *A*, cujos elementos $A_{ij} = 1$ se existe uma ligação entre *i* e *j*, senão $A_{ij} = 0$. O grau, isto é, o número de vizinhos conectados, de um vértice *i* é denotado por K_i e dado por $A_{ij} = \sum_j A_{ij}$.

Um caminhante no vértice *i* e no tempo *t* seleciona um de seus K_i vizinhos com igual probabilidade no tempo $t + 1$, a probabilidade de transição do vértice *i* para o vértice *j* é A_{ij}/K_i. Supõe-se que o caminhante comece no vértice *i* no tempo *t* = 0, desse modo a equação para a probabilidade P_{ij} para encontrar o caminhante no vértice *j* no tempo *t* é:

$$P_{ij}(t+1) = \sum_k \frac{A_{kj}}{K_k} P_{ik}(t) \tag{6}$$

O valor máximo do autovalor correspondente a evolução no tempo do operador é 1 correspondendo a distribuição estacionária $P_j^{\infty} = lim_{t\to\infty} P_{ij}(t)$, isto é, o limite de tempo infinito. Uma expressão para a probabilidade de transição $P_{ij}(t)$ para ir do vértice *i* para o vértice *j* em *t* passos segue iterando-se a Equação (6).

$$P_{ij}(t) = \sum_{j_1,\cdots,j_{t-1}} \frac{A_{ij_1}}{K_i}, \frac{A_{j_1 j_2}}{K_{j_1}} \cdots \frac{A_{j_{t-1} j}}{K_{j_{t-1}}} \tag{7}$$

Comparando as expressões para P_{ij} e P_{ji} nota-se que:

$$K_i P_{ij}(t) = K_j P_{ji}(t) \tag{8}$$

Isto é uma conseqüência direta do não direcionamento da rede. Para a solução estacionária, a Equação (8) implica que $K_i P_j^{\infty} = K_j P_i^{\infty}$, e daí obtém-se:

$$P_i^{\infty} = \frac{K_i}{N} \tag{9}$$

onde $N = \sum_i K_i$. Note que a distribuição estacionária é igual ao grau do vértice *i*, quanto mais ligações um vértice possuir com outros vértices na rede, mais frequentemente ele será visitado por um caminhante aleatório.

A probabilidade da primeira passagem $F_{ij}(t)$ do vértice *i* ao vértice *j* após *t* passos satisfaz a relação:

$$P_{ij}(t) = \delta_{t0}\delta_{ij} + \sum_{t'=0}^{t} P_{jj}(t - t')\, F_{ij}(t') \tag{10}$$

O símbolo delta de Kronecker garante a condição inicial $P_{ij}(0) = \delta_{ij}$, ($F_{ij}(0)$ é setado com zero). Introduzindo a transformada Laplaciana $\tilde{f}(s) \equiv \sum_{t=0}^{\infty} e^{-st} f(t)$, a Equação (10) torna-se $\tilde{P}_{ij}(s) = \delta_{ij} + \tilde{F}_{ij}(s)\tilde{P}_{jj}(s)$ e se obtém:

$$\tilde{F}_{ij}(s) = (\tilde{P}_{ij}(s) - \delta_{ij})/\tilde{P}_{jj}(s) \tag{11}$$

Em redes finitas a caminhada aleatória é recorrente, então a média de tempo da primeira passagem é dada por $\langle T_{ij} \rangle = \sum_{t=0}^{\infty} t F_{ij}(t) = -\tilde{F}'_{ij}(0)$.

Desde que todos os momentos $R_{ij}{}^{(n)} \equiv \sum_{t=0}^{\infty} t^n \{P_{ij}(t) - P_j^{\infty})$ da relaxação exponencial do decaimento que partem de $P_{ij}(t)$ são finitos, pode-se expandir $\tilde{P}_{ij}$ como uma série em *s:*

$$\tilde{P}_{ij}(s) = \frac{K_j}{N(1-e^{-s})} + \sum_{n=0}^{\infty} (-1)^n R_{ij}^{(n)} \frac{s^n}{n!} \tag{12}$$

Inserindo esta série na Equação (11) e expandindo-a como uma série de potência em *s*, obtém-se:

$$\langle T_{ij} \rangle = \begin{cases} \frac{N}{K_j}, & para\ j = i \\ \frac{N}{K_j}\left[R_{jj}^{(0)} - R_{ij}^{(0)}\right], & para\ j \neq i \end{cases} \tag{13}$$

A caminhada aleatória entre dois vértices é assimétrica. A diferença entre $\langle T_{ij} \rangle$ e $\langle T_{ji} \rangle$ para $j \neq i$ pode ser escrita como (usando a Equação 10):

$$\langle T_{ij} \rangle - \langle T_{ji} \rangle = N\left(\frac{R_{jj}^{(0)}}{K_j} - \frac{R_{ii}^{(0)}}{K_i}\right) - N\left(\frac{R_{ij}^{(0)}}{K_j} - \frac{R_{ji}^{(0)}}{K_i}\right) \tag{14}$$

onde o último termo desaparece pela Equação (11). Assim, obtém-se:

$$\langle T_{ij} \rangle - \langle T_{ji} \rangle = C_j^{-1} - C_i^{-1} \tag{15}$$

$$C_i = \frac{P_i^{\infty}}{\tau_i} \tag{16}$$

onde $P_i^{\infty} = K_i/N$ e o tempo característico de relaxação τ_i do vértice *i* é dado por:

$$\tau_i = R_{ii}^{(0)} = \sum_{t=0}^{\infty}\{P_{ii}(t) - P_i^{\infty}\} \tag{17}$$

Coeficiente de *clustering:* o *coeficiente de clustering* proposto por Watts e Strogatz (1998), caracteriza a "densidade" de conexões em torno de um vértice. Supondo que um vértice *i* em uma rede tem *k* vizinhos mais próximos. Isto significa que existem $k(k-1)/2$ possíveis arestas entre os vizinhos mais próximos de i. O coeficiente de *clustering* CC_i de um vértice é a razão entre o número total de arestas *y* conectando seus vizinhos mais próximos e o número total de arestas possíveis entre estes vizinhos mais próximos.

$$CC_i = \frac{2y}{k(k-1)} \tag{18}$$

Normalmente a média $\langle CC \rangle$ é considerada e representa o coeficiente de *clustering* da rede. $\langle CC \rangle$ é a probabilidade de que se uma tripla de vértices de uma rede é conectada por pelo menos duas arestas, então a terceira aresta também está presente. Essa medida é uma forma específica de indicar existência de estruturas locais em uma rede.

$$\langle CC \rangle = \sum_{i=1}^{N} C_i \tag{19}$$

2.2 Tipos de redes

Nesta seção são descritos os três principais modelos de redes complexas: aleatória, mundo pequeno e livre de escala.

2.2.1 Redes aleatórias

O estudo das redes aleatórias foi iniciado por Erdös e Rényi (1959), quando eles estudavam por meios de métodos probabilísticos propriedades dos grafos em função do aumento do número de conexões aleatórias. Eles propuseram um modelo para gerar grafos aleatórios com *N* vértices e *k* arestas. Iniciando-se com *N* vértices desconectados, pares de vértices são selecionados aleatoriamente, evitando-se múltiplas conexões, até que o número de arestas seja igual a *k*. Um modelo alternativo consiste em conectar os pares de vértices com uma probabilidade $0 < p < 1$. Este procedimento resulta em grafos com diferentes números de arestas. A probabilidade de um grafo com *k* arestas é $p^k(1-p)^{N(N-1)/2-k}$. A probabilidade que um vértice *i* tenha *k* arestas é dada pela distribuição binomial:

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{20}$$

na qual p^k é a probabilidade de existirem *k* arestas, $(1-p)^{N-1-k}$ é a probabilidade de ausência das $N - 1 - k$ arestas restantes e $\binom{N-1}{k}$ é o número de maneiras diferentes de selecionar os pontos finais das *k* arestas. Como todos os vértices numa rede aleatória são estatisticamente equivalentes, todos tem a mesma distribuição, e a probabilidade de que um vértice escolhido aleatoriamente tenha grau *k* é *P*(*k*). Para *N* grande, e $\langle k \rangle$ fixo, a distribuição do grau é aproximada a distribuição de *Poisson*:

$$P(k) = \frac{e^{-\langle k \rangle} \langle k \rangle^k}{k!} \tag{21}$$

2.2.2 Redes mundo pequeno

Na maioria das redes reais, apesar do tamanho grande, há um caminho relativamente curto entre quaisquer pares de vértices. Essa característica é conhecida como propriedade de mundo pequeno (*small world*) e foi investigada primeiramente por Milgran (1967). Em um de seus experimentos, Milgran selecionou aleatoriamente algumas pessoas da cidade de Omaha e pediu para enviarem cartas para indivíduos distantes da cidade de Boston, identificados apenas por seus nomes, ocupação e endereço. As cartas apenas deveriam ser enviadas para alguém que as pessoas conhecessem e que se presumia estarem perto do destinatário final. Ele

documentou que em uma rede social, a distância média entre duas pessoas é muito pequena, em média seis passos são suficientes para conectar duas pessoas quaisquer.

Watts e Strogatz (1998) notaram que muitas redes reais possuíam esta característica e um alto valor para o coeficiente de *clustering*. Eles propuseram então um método de construção dessas redes, o qual é baseado em um processo de re-ligação das arestas por meio de uma probabilidade p. O ponto inicial é um anel com N vértices, no qual cada vértice é simetricamente ligado aos seus $2k$ vizinhos mais próximos para um total de arestas $K = kN$. Para cada vértice, cada aresta ligada a um vizinho no sentido horário, é re-ligada a um vértice, escolhido aleatoriamente, com uma probabilidade p, e preservada, com uma probabilidade $1 - p$. Nota-se que para $p = 1$, o modelo produz uma rede aleatória com a restrição de que cada vértice tem conectividade mínima $k_{min} = k$. Para valores pequenos de p o procedimento gera redes com propriedade de mundo pequeno e coeficiente de *clustering* não trivial. A Figura 1 ilustra o modelo proposto por Watts e Strogatz.

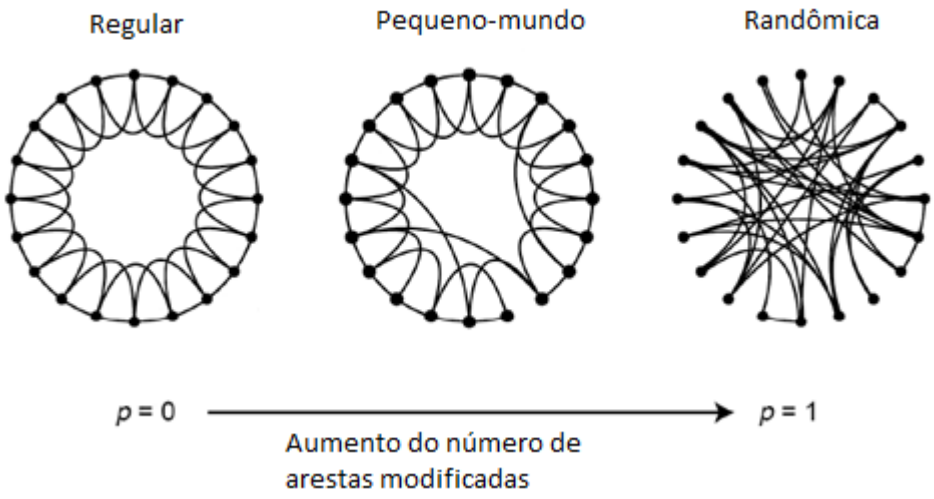


Figura 1: O modelo de Watts-Strogatz para redes mundo pequeno. Adaptada de Watts e Strogatz (1998).

2.2.3 Redes livre de escala

Conforme os cientistas abordavam o estudo das redes reais a partir das bases de dados disponíveis, verificavam que a maioria das redes não possuíam uma distribuição de grau regular, mas seguiam a distribuição da lei de potência, $P(k) \sim k^{-\gamma}$, com expoentes variando de $2 < \gamma < 3$. Essas redes foram nomeadas redes livres de escala devido à lei de potência e possuem uma distribuição de grau altamente heterogênea, com alguns vértices de grau elevado (*hubs*) ligados a muitos outros de graus baixos. Um exemplo é a rede de colaboração entre cientistas (Figura 2).

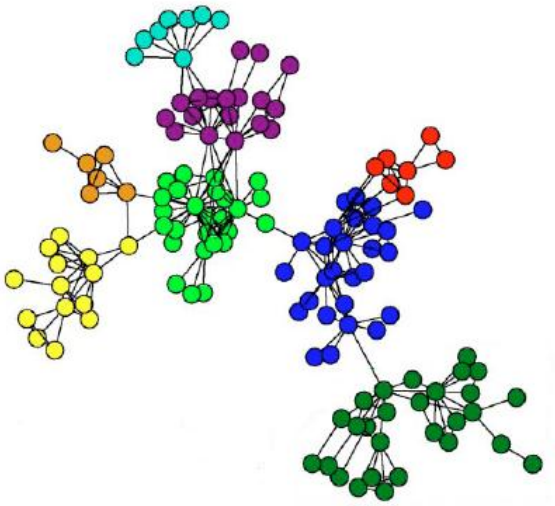


Figura 2: Rede de colaboração entre cientistas. Obtida em Newman (2004).

Barabási e Albert (1999) propuseram um modelo para esse tipo de rede, inspirados na formação da WWW, o qual é baseado principalmente em crescimento e ligações preferenciais. A idéia básica é que na WWW, sites com alto grau, adquirem novos vértices em taxas maiores que vértices com baixo grau. A Figura 3 mostra a distribuição de grau calculada pelos autores para algumas redes reais.

Iniciando com m_0 vértices isolados, a cada passo de tempo t = 1, 2, 3 ... $N - m_0$ um novo vértice j com m arestas é adicionado à rede. A probabilidade de um novo vértice j ser conectado a um vértice k já existente é dada pela Equação (22):

$$P(j \to k) = \frac{g_k}{\sum_l g_l} \tag{22}$$

onde g_k é o grau do vértice k. Os vértices com mais ligações na rede tendem a ser preferidos nas ligações. Esses vértices tendem a ser altamente conectados e iterativamente tornam-se ainda mais conectados, efeito conhecido como o "rico torna-se mais rico" (Barabási, 2003).

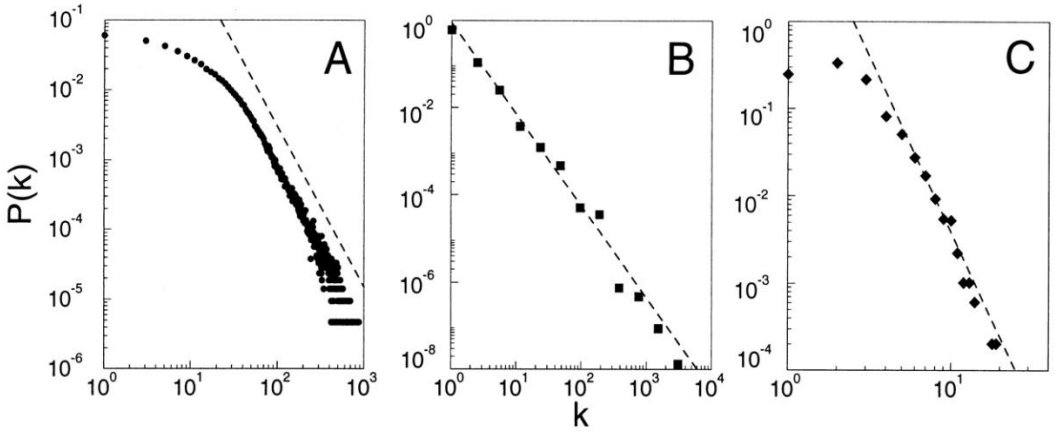


Figura 3: Distribuição do grau para várias redes reais. (A) Rede de colaboração entre atores, N = 212250 e $\langle g \rangle$ = 28.78. (B)WWW, N = 325729, $\langle g \rangle$ = 5,46. (C) Rede elétrica, N = 4941 e $\langle g \rangle$ = 2.67. Obtida em Barabási e Albert (1999).

2.3 Métodos de detecção de comunidades em redes complexas

Outra propriedade importante é a existência de comunidades na rede. Newman (2003) define comunidade como grupos de vértices que são fortemente conectados entre si e fracamente conectados com elementos de outro grupo. Existem diversos métodos na literatura para efetuar a divisão de comunidades, a seguir são apresentados alguns métodos mais importantes.

2.3.1 *Betweenness*

Nesse método, proposto por Newman e Girvan (2004), as arestas com maior *betweenness* são sucessivamente removidas e a rede é decomposta em sub-redes separadas. Como em geral existem mais arestas ligando pares de vértices dentro das comunidades do que arestas interligando comunidades, qualquer caminho entre vértices de comunidades diferentes passarão por alguma dessas poucas arestas que as interligam. Como o *betweenness* dessas arestas tende a ser maior, ao eliminá-las obtêm-se comunidades distintas.

Os autores apresentam várias implementações dessa ideia:

i) Para calcular o *betweenness* baseado no caminho geodésico para todos os pares de vértices, os autores sugerem um algoritmo mais rápido, que usa busca em profundidade. Esse algoritmo leva $O(mn^2)$ operações em um grafo com m arestas e n vértices. Newman (2001) e Brandes (2001) também propuseram novos algoritmos que fazem o cálculo de maneira ainda mais rápida, encontrando todos os *betweennesses* em tempo $O(mn)$.

ii) Outra medida chamada *random-walk betweenness* é baseada em sinais caminhando aleatoriamente pela rede. É possível estimar o número de vezes que um sinal passa por uma determinada aresta para fazer um caminho aleatório entre um par de vértices.

iii) Outra medida é motivada por ideias de teoria elementar de circuitos. O circuito é criado colocando-se um resistor em cada aresta da rede e uma corrente com fonte e coletor colocados em um determinado par de vértices. O fluxo de corrente percorrerá a rede da fonte ao coletor por muitos caminhos, sendo que aqueles com menor resistência carregarão a maior fração da corrente. Assim a medida de *betweeness* de uma aresta é definida como sendo o valor absoluto da soma da corrente que passa pela aresta com todos os pares de fonte/coletor.

2.3.2 Modularidade

Newman (2004) propõe um algoritmo baseado na modularidade (Newman e Girvan, 2004), a qual mede a qualidade de uma divisão da rede. Uma divisão é considerada boa se possuir muitas arestas intra comunidades e poucas arestas entre comunidades.

Dada uma rede com *n* comunidades, a modularidade Q é calculada por uma matriz simétrica de *n* linhas e *n* colunas, na qual os elementos ao longo da diagonal principal, e_{ii}, representam as conexões entre os vértices na mesma comunidade e os elementos e_{ij}, representam as conexões entre as comunidades i e j. A Equação (23) traz a medida de modularidade Q.

$$Q = \sum_i \left[e_{ii} - \left(\sum_j e_{ij} \right)^2 \right] \tag{23}$$

Essa equação mede a fração de arestas na rede que conectam vértices de uma mesma comunidade, menos o valor esperado da mesma divisão, mas considerando-se conexões aleatórias entre os vértices. Se uma divisão particular fornece menos arestas entre comunidades do que seria esperado por conexões aleatórias, essa modularidade é $Q = 0$. Valores maiores que 0 indicam desvios da aleatoriedade, tal que, Q próximo de 1 indica que a rede é formada por estruturas modulares bem definidas. Porém, na prática, valores de $Q \geq 0.3$ já indicam uma boa divisão.

Por meio dessa medida, Newman (2004) propôs um algoritmo de otimização guloso, categorizado como um método aglomerativo, no qual uma rede de *n* vértices começa sem nenhuma conexão, tendo inicialmente *n* comunidades. A cada iteração duas comunidades c_i e c_j, que tenham conexão na rede real, são conectadas se fornecerem o maior acréscimo no valor da Equação (24).

$$\Delta Q_{ij} = 2\left(e_{ij} - \sum_j e_{ij} \ \ \sum_i e_{ji}\right) \tag{24}$$

Para calcular os valores de ΔQ, é necessário considerar apenas os pares de vértices entre os quais existem arestas, já que pares entre os quais não há nenhuma aresta nunca resultariam em um acréscimo em ΔQ, com isso o tempo de execução do algoritmo é reduzido.

O resultado pode ser representado como um dendograma (Figura 4). Cortes em diferentes níveis do dendograma darão divisões em maior ou menor número de comunidades, sendo que o melhor corte seria aquele dado pelo maior valor de Q.

O algoritmo em cada passo possui complexidade $O(m + n)$. Como há no máximo *n* - 1 operações de junções necessárias para construir um dendograma completo, sua complexidade total é $O\big((m + n)n\big)$, ou $O(n^2)$ em um grafo esparso.

Posteriormente, Clauset, Newman e Moore (2004) apresentam um novo algoritmo que funciona em tempo $O(n\,log^2 n)$ para uma rede esparsa com *n* vértices, permitindo assim a análise de comunidades em redes consideradas muito grandes para serem tratadas.

Newman (2004) destaca que esse método de otimização é mais rápido que o *Betweenness*, porém *Betweenness* produz resultados melhores, sendo mais recomendado em redes menores.

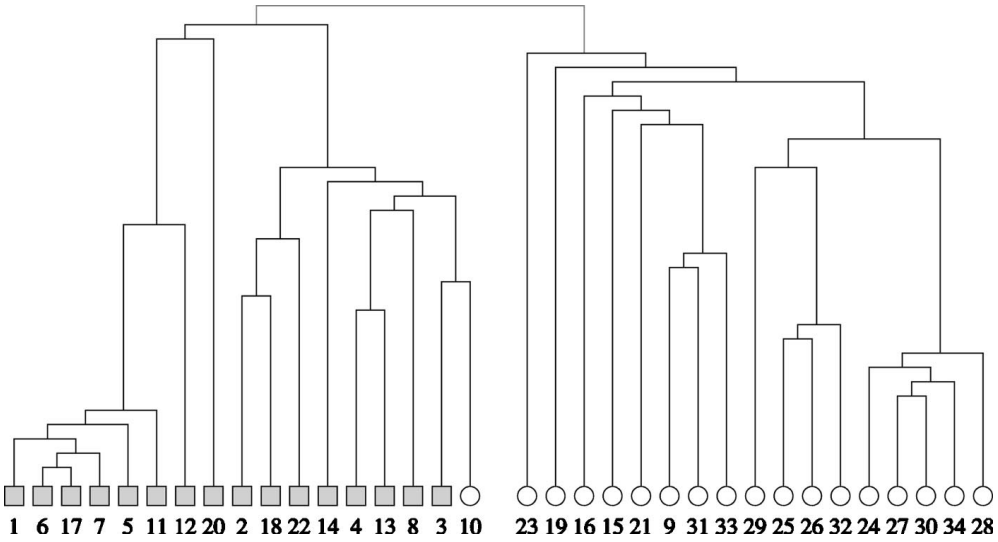


Figura 4: Dendograma das comunidades encontradas pelo algoritmo de modularidade Q (Newman, 2004) para a rede clube de karatê. As formas dos vértices representam os dois grupos nos quais o clube se separou. Figura obtida em Newman (2004).

2.3.3 Caminhada aleatória

Zhou (2003a, 2003b) utiliza o conceito de movimento browniano em uma rede para detecção de comunidades. Ele sugere que se uma partícula Browniana passeia em uma rede por certo tempo, essa obtém uma perspectiva do panorama da rede. A distância entre os vértices medida por esta partícula pode ser usada para construir a estrutura da comunidade e identificar o vértice central da comunidade.

Em Zhou (2003a) é apresentado um método baseado no conceito de atratores locais e atratores globais, e em Zhou (2003b) é definida uma medida chamada *índice de dissimilaridade* entre vizinhos mais próximos, a qual indica a probabilidade de dois vértices estarem na mesma comunidade. A seguir esta medida é explicada com mais detalhes.

Seja uma rede conectada de *N* vértices e *M* arestas, com conexões representadas pela matriz de adjacência *A*. Uma partícula Browniana se mantém em movimento na rede e a cada passo ela pula de uma posição *i* para um vizinho próximo *j*, essa probabilidade é dada pela Equação (25), e a matriz correspondente *P* é chamada de matriz de transferência.

$$P_{ij} = \frac{A_{ij}}{\sum_{l=1}^{N} A_{il}} \tag{25}$$

A distância $d_{i,j}$ do vértice *i* ao vértice *j* é definida como a média do número de passos necessários para que a partícula Browniana locomova-se do vértice *i* ao vértice *j*. Pode ser calculada através da solução algébrica da Equação (26):

$$d_{i,j} = \sum_{l=1}^{N} \left(\frac{1}{I - B(j)}\right)_{il}, \tag{26}$$

na qual *I* é a matriz identidade $N\chi N$ e $B(j)$ é a matriz de transferência *P* exceto que $B_{lj}(j) = 0, \forall l \in V$. A distância de todos os vértices da rede a *j* pode ser obtida a partir da resolução da Equação (27).

$$[I - B(j)]\{d_{1,j}, d_{2,j}, \cdots, d_{N,j}\}^T = \{1, \cdots, 1\}^T \tag{27}$$

Dado um vértice *j*, se $d_{i,j} \leq d_{i,m}, \forall m \in V$, *j* é dito atrator global de *i*. Se *j* for vizinho de *i* e $d_{i,j} \leq d_{i,m}$ para todo *m* vizinho de *i*, então diz-se que *j* é um atrator local de *i*. Dada uma rede com comunidades bem definidas, um vértice *i* tem uma alta probabilidade de pertencer a mesma comunidade que seu atrator local *j*, já que dentre todos os vértices vizinhos de *i*, *j* apresenta a menor distância $d_{i,j}$. Desse modo, dado um vértice *i* participante de uma comunidade *L*, um vértice *j* é acrescentado a comunidade *L* se *j* for um atrator local de *i* ou se *i* for um atrator local de *j*.

Em Zhou (2003b) esse método é estendido e é criado um *índice de dissimilaridade* entre pares de vértices. Um algoritmo hierárquico faz uso desses índices para decompor a rede em uma sequência de grupos. Considera-se que um vértice deve ter maior interação com vértices de sua própria comunidade do que com vértices de outra comunidade do grafo.

Tomando-se qualquer vértice *i* como origem da rede, o conjunto $\{d_{i,1}, \cdots, d_{i,i-1}, d_{i,i+1}, \cdots, d_{i,N}\}$ mede a distância de todos os outros vértices com relação a origem, ou seja, representa uma visão de toda a rede do ponto de vista de *i*. Dado dois vértices *i* e *j* que são vizinhos mais próximos, a diferença entre seus pontos de vista pode ser medida quantitativamente através do *índice de dissimilaridade*, calculado pela Equação (28):

$$\Lambda(i,j) = \frac{\sqrt{\sum_{k \neq i,j}^{N} [d_{ik} - d_{jk}]^2}}{N-2} \tag{28}$$

Se dois vértices *i* e *j* são vizinhos mais próximos que pertencem à mesma comunidade, então as distâncias d_{ik} e d_{jk}, onde *k* é qualquer outro ponto do grafo (com $k \neq i, j$), serão bastante similares, assim a perspectiva da rede do ponto de vista de *i* e *j* também serão

bastante similares. Consequentemente, $\Lambda(i,j)$ será pequeno se i e j pertencerem a mesma comunidade e grande se pertencerem a comunidades diferentes.

2.3.4 Competição de partículas

Quiles et al. (2008) apresentam um modelo de detecção de comunidades dinâmico, baseado em caminhada aleatória e competição de partículas. Nesse modelo, várias partículas são colocadas para caminhar em uma rede, competindo entre si para marcar vértices sob seu domínio e rejeitar partículas intrusas. Cada partícula possui uma variável que representa o seu potencial de exploração, denominado potencial da partícula, e cada vértice possui uma variável que representa quão intenso é o domínio realizado por sua partícula, denominado potencial do vértice.

O potencial da partícula aumenta quando ela visita seus próprios vértices e diminui quando ela choca com um vértice pertencente a outra partícula. Da mesma maneira, quando um vértice é visitado por sua própria partícula, ou está sendo visitado pela primeira vez, seu potencial aumenta; e quando ele é visitado por uma partícula intrusa, um choque acontece e o potencial do vértice é decrementado. Com isso, um vértice pode mudar de dono se ele for visitado primeiramente por uma partícula, mas frequentemente passa a ser visitado por outras partículas.

O modelo é descrito como se segue: no início, um conjunto *K* de partículas são inseridas aleatoriamente em uma rede. Essas partículas caminham na rede tomando seus próprios vértices. Há dois tipos de dinâmica: a dinâmica das partículas e a dinâmica dos vértices. Cada partícula ρ_j possuí duas variáveis $\rho_j^v(t)$ e $\rho_j^\omega(t)$, sendo que $\rho_j^v(t)$ é usado para representar o vértice v_i sendo visitado por uma partícula ρ_j no tempo t e $\rho_j^w(t) \in [\omega_{min}, \omega_{max}]$ é o potencial da partícula caracterizando a habilidade de exploração da partícula j no tempo t. Especificamente, a dinâmica da partícula é governada pelas Equações (29) e (30):

$$\rho_j^v(t+1) = v_i, \tag{29}$$

$$\rho_j^w(t+1) = \begin{cases} \rho_j^\omega(t), & se\ v_i^p(t) = 0, \\ \rho_j^\omega(t) + \left(\omega_{max} - \rho_j^\omega(t)\right)\Delta_\rho & se\ v_i^p(t) = \rho_j \neq 0, \\ \rho_j^\omega(t) - \left(\rho_j^\omega(t) - \omega_{min}\right)\Delta_\rho & se\ v_i^p(t) \neq \rho_j \neq 0, \end{cases} \tag{30}$$

na qual $0 < \Delta_\rho \leq 1$ é um parâmetro que controla a velocidade do aumento ou diminuição do potencial da partícula. ω_{max} e ω_{min} representa o maior e o menor valor permitido para o potencial de todas as partículas, respectivamente.

Cada vértice v_i possui três variáveis: $v_i^p(t)$, $v_i^\omega(t)$ e v_i^γ. A primeira registra a partícula "dona" do vértice v_i no tempo *t*; ela leva o valor ρ_j se está ocupada por uma partícula ρ_j ou 0 se o vértice v_i estiver em um estado livre (o vértice ainda não foi dominado por nenhuma partícula). A segunda variável, $v_i^\omega(t)$, assim como $p_i^\omega(t)$ para uma partícula, é o potencial de um vértice v_i no tempo *t* e representa a força da dominância da partícula ρ_j sobre o vértice v_i, isto é, um alto valor de $v_i^\omega(t)$ significa que v_i é fortemente dominada por ρ_j, um baixo valor significa uma dominância fraca, e quando $v_i^\omega(t) = \omega_{min}$ indica que o vértice v_i está em estado livre. A terceira variável v_i^γ é uma variável binária que assume o valor 0 se o vértice v_i não está sendo visitado por alguma partícula nesse momento, e assume o valor 1 caso esteja sendo visitado por alguma partícula. As Equações (31) e (32) descrevem a dinâmica do vértice:

$$v_i^\rho(t+1) = \begin{cases} v_i^\rho(t) & se\ v_i^\gamma = 0 \\ \rho_j & se\ v_i^\gamma = 1\ e\ \ v_i^\omega = \omega_{min} \end{cases}, \tag{31}$$

$$v_i^\omega(t+1) = \begin{cases} v_i^\omega(t) & se\ v_i^\gamma = 0\,, \\ max\{\omega_{min},\ v_i^\omega(t) - \Delta_v\} & se\ \ v_i^\gamma = 1\ e\ \ v_i^\rho \neq \rho_j \\ \rho_j^\omega(t+1) & se\ \ v_i^\gamma = 1\ e\ \ v_i^\rho(t) = \rho_j, \end{cases}, \tag{32}$$

na qual $0 < \Delta_\rho \leq 1$ é um parâmetro que controla a velocidade de mudança do potencial dos vértices. Se $\Delta_\rho(\Delta_v)$ é um valor baixo, o potencial da partícula muda devagar, caso contrário, se o valor for alto, o potencial da partícula muda rapidamente.

O processo de detecção de comunidades aplicando o modelo pode ser descrito como segue. Inicialmente *K* partículas são colocadas em *K* vértices escolhidos aleatoriamente. Cada partícula ρ_j tem seu potencial inicial $\rho_j^\omega(0) = \omega_{min}$ e cada vértice v_i tem seu potencial inicial $v_i^\omega(t) = \omega_{min}$ também. Até esse momento, todos os vértices estão livres, isto é, $v_i^p(0) = 0$. Conforme o sistema executa, cada partícula escolhe um vizinho para visitar em cada iteração. A partícula encontra uma das seguintes situações em cada visita:

1. Se o vértice v_i está sendo visitado por uma partícula ρ_j e não tem nenhum dono ainda, o potencial de ρ_j não muda, mas o dono de v_i é marcado como sendo ρ_j, isto é, $v_i^p(t) = \rho_j$, e o potencial de v_i recebe o potencial de ρ_j, isto é, $v_i^\omega(t) = \rho_j^\omega(t)$.

2. Se o vértice que está sendo visitado pela partícula ρ_j pertence a essa partícula, isto é, se $v_i^p(t) = \rho_j$, o potencial de ρ_j é aumentado aplicando-se a segunda linha da Equação (30), a dominância de v_i é mantida como sendo ρ_j, e novamente o potencial de v_i recebe o potencial de ρ_j.
3. Se o vértice v_i que está sendo visitado por uma partícula ρ_j, pertence a outra partícula, um choque ocorre e a partícula ρ_j é rejeitada pelo vértice v_i. Nesse caso, o potencial de ambos é decrementado aplicando-se a terceira linha da Equação (30) e a segunda linha da Equação (32), respectivamente. Se $\rho_j^{\omega}(t)$ é reduzido abaixo de ω_{min}, ele é escolhido por outra partícula aleatoriamente, e seu potencial se torna ω_{min}, sua dominância é 0, indicando que o vértice pode ser ocupado por qualquer partícula.

Dessa maneira, a dominância de um vértice é fortalecida se o mesmo é visitado pela mesma partícula frequentemente e é reduzido ou até mesmo mudado se ele é frequentemente visitado por outras partículas diferente de sua dona. Este processo continua até que um estado de equilíbrio dinâmico ser encontrado.

A fim de estudar a combinação do movimento aleatório e determinístico, é definida uma probabilidade $0 \leq p_{det} \leq 1$. A cada iteração, cada partícula tem uma probabilidade p_{det} de tomar movimentos determinísticos e probabilidade $1 - p_{det}$ de tomar movimentos aleatórios.

2.4 Otimização em redes complexas

Cancho e Solé (2003) utilizam um algoritmo evolutivo envolvendo minimização da quantidade de arestas e da média do menor caminho e encontraram quatro principais tipos de redes: exponenciais, livres de escala, estrelas e altamente densas. A seguir esse trabalho é descrito com mais detalhes.

Seja um grafo não-direcionado com número fixo de vértices e arestas, definido por uma matriz de adjacência $A = \{a_{ij}\}, 1 \leq i, j \leq n$ e seja D_{ij} a distância mínima entre eles. Em um tempo t = 0, tem-se um grafo aleatório, com distribuição de grau de Poisson, no qual dois vértices são conectados com uma probabilidade p. A função de energia do algoritmo de otimização é definida como:

$$E(\lambda) = \lambda d + (1 - \lambda)\rho \tag{33}$$

na qual $0 \leq \lambda, d, \rho \leq 1$. λ é um parâmetro que controla a combinação linear de d e ρ. O número normalizado de arestas, ρ é definido em termos de a_{ij} como:

$$\rho = \frac{1}{\binom{n}{2}} \sum_{i<j} a_{ij} \tag{34}$$

A distância vértice-vértice normalizada, d, é definida como $d = D/D^{linear}$, tal que D é a média da distância mínima entre os vértices e $D^{linear} = (n+1)/3$ é o valor máximo de D que pode ser encontrado em uma rede conexa, considerando-se um grafo linear.

$$D = \frac{1}{\binom{n}{2}} \sum_{i<j} D_{ij} \tag{35}$$

A minimização de $E(\lambda)$ envolve a minimização simultânea da distância e do número de arestas. Essas duas restrições incluem dois aspectos relevantes em uma rede, o custo das ligações físicas entre unidades e a velocidade de comunicação entre elas.

O algoritmo de minimização procede da seguinte maneira: No tempo $t = 0$, a rede é inicializada com uma densidade $\rho(0)$ seguindo a distribuição de grau de Poisson. Num tempo $t > 0$, o grafo é modificado alterando-se aleatoriamente a conexão entre alguns pares de vértices. Com uma probabilidade v, cada a_{ij} pode mudar de 0 para 1 ou de 1 para 0. A nova matriz de adjacência é aceita se $E(\lambda, t+1) < E(\lambda, t)$. Caso contrário, diferentes alterações são realizadas e testadas novamente. O algoritmo é interrompido quando as modificações em $A(t)$ não são aceitas após T execuções. A minimização do algoritmo funciona como a técnica de *simulated annealing* na temperatura zero.

A Figura 5 mostra algumas das propriedades básicas mostradas pelas redes otimizadas. Tais propriedades juntamente com a Figura 6, sugerem que quatro fases estão presentes, separadas por três transições em $\lambda_1 \approx 0.25$, $\lambda_2 \approx 0.80$ e $\lambda_3 \approx 0.95$.

Uma análise mais detalhada da transição entre as distribuições de grau revela que a formação de *hubs* explica o surgimento de (B) a partir de (A), a competição de *hubs* (B') precede o aparecimento de um vértice central (C). O surgimento de grafos densos em (C) consiste de um aumento progressivo do grau médio de vértices não centrais e uma súbita perda de seu vértice central. A Figura 6 mostra $\langle H(\lambda) \rangle$ juntamente com imagens dos principais tipos de redes, pode ser visto que as redes sem escala (B) são encontradas perto de λ_1.

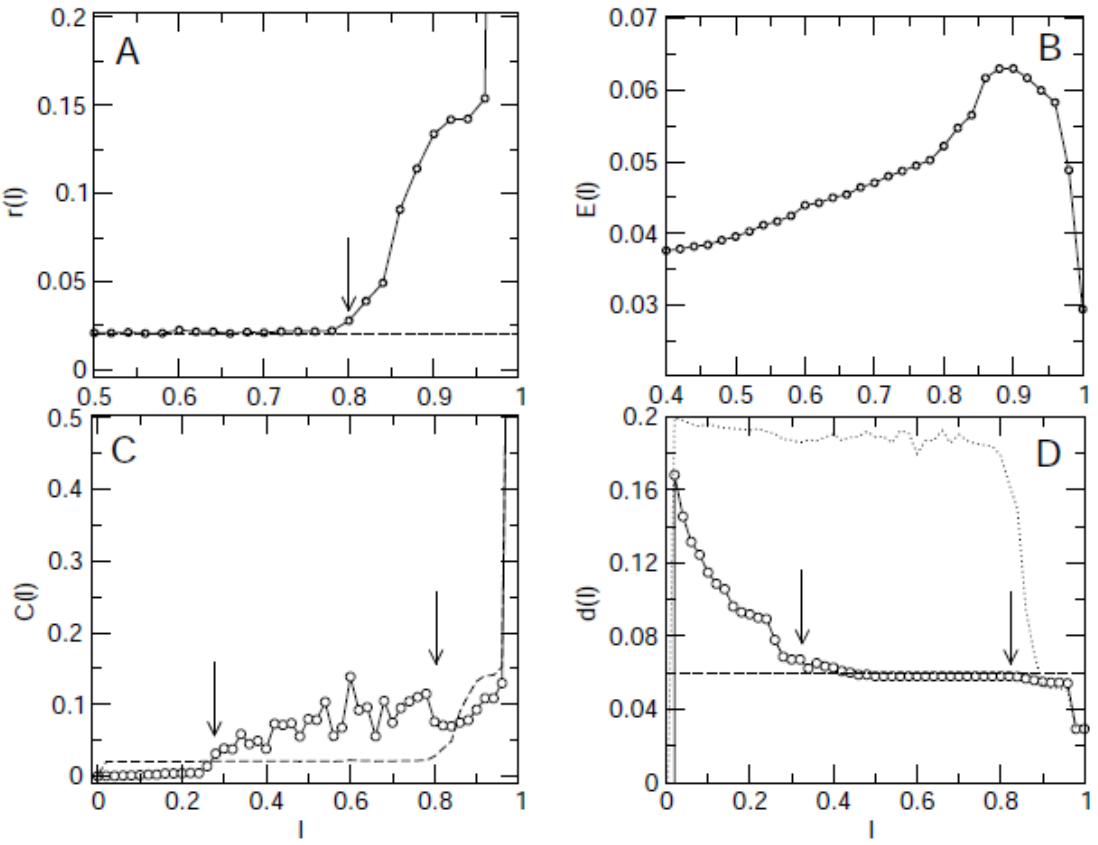


Figura 5: (A) Densidade de *links*, (B) energia, (C) coeficiente de *clustering* e (D) distância como uma função de λ. Média sobre 50 redes otimizadas com n = 100, $T = \binom{n}{2}$, $v = 2/\binom{n}{2}$ e $\rho(0) = 0.2$ são mostradas. A: a rede ótima se torna um grafo completo para λ perto de 1. A densidade para uma rede estrela, $p_{star} = 2/n = 0.02$ é mostrada como referência (linha tracejada). O coeficiente de *clustering* de uma rede de Poisson $C_{random} = \langle k \rangle/(n-1)$ é mostrado como referência em C. A distância normalizada de uma rede estrela, $d_{star} = 6(n-1)/(n(n+1) = 0.058$ (linha tracejada) e de uma rede de Poisson, $d_{random} = \log n \,/\, \log\langle k \rangle$ (linha pontilhada) são mostradas como referência em D. Figura obtida em Cancho e Solé (2003).

O trabalho de Cancho e Solé (2003) mostra que a otimização tem um papel fundamental na formação e evolução das redes complexas. Como as redes estrelas não são observadas em sistemas reais, os autores apresentam algumas restrições que possivelmente impedem este fato:

- aleatoriedade: a evolução da topologia conforme λ aumenta sugere uma transição de desordem (distribuição de grau exponencial) para ordem (distribuição de grau estrela);
- diversidade: o número de diferentes redes estrelas que podem ser formadas com n vértices é n e isso aumenta para distribuições exponenciais;
- robustez: removendo o *hub* central restam n – 1 componentes desconexos, o que seria o pior caso.

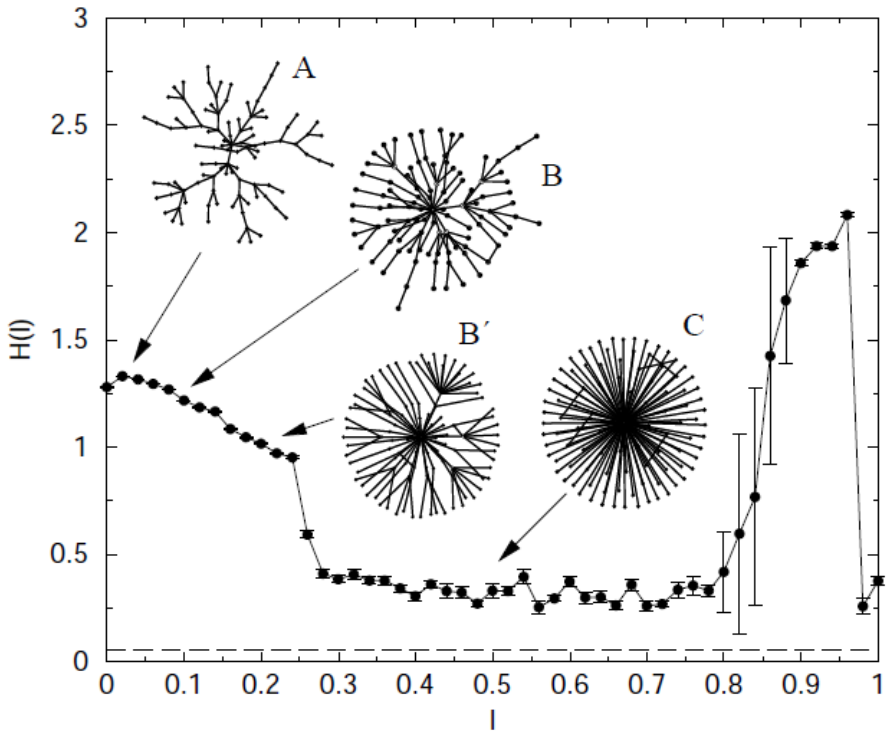


Figura 6: Redes ótimas para determinados valores de λ são mostradas. Média sobre 50 redes otimizadas com n = 100, $T = \binom{n}{2}$, $v = 2/\binom{n}{2}$ e $\rho(0) = 0.2$. A: uma rede exponencial com $\lambda = 0.01$. B: uma rede livre de escala com $\lambda = 0.08$. *Hubs* com múltiplas conexões e a dominância de vértices com apenas uma conexão podem ser vistos. C: uma rede estrela com $\lambda = 0.5$. B': uma rede intermediária entre B e C com muitos *hubs* podem ser identificadas. A entropia de uma rede estrela, $H_{estrela} = \log n - \left[\frac{n-1}{n}\right]\log(n-1) = 0.056$ é mostrada como referência. Figura obtida em Cancho e Solé (2003).

Capítulo 3

Revisão sobre detecção de *outliers* e classificação de dados

O aprendizado de máquina é um sub-campo da Inteligência Artificial e tem apresentado um grande progresso nos últimos anos, possuindo aplicações em diversas áreas como a indústria, medicina, economia, ecologia, finanças, entre outras. O princípio básico do aprendizado de máquina é o desenvolvimento de algoritmos que melhoram seus desempenhos automaticamente com a experiência (Mitchell, 1997).

No que segue a Seção 3.1 expõe alguns conceitos básicos de aprendizado de máquina. A Seção 3.2 descreve sobre detecção de *outlier* e a Seção 3.3 trata da classificação de dados, apresentando brevemente algumas técnicas de classificação e a técnica baseada em redes *K*-associados.

3.1 Aprendizado de máquina

O aprendizado de máquina busca propor e desenvolver técnicas que permitem aos computadores "aprender" ou melhorar seu desempenho por meio da experiência (Mitchell, 1997). Desse modo, procura aprender a extrair automaticamente padrões complexos em bases de dados transformando-os em informações que possam ser utilizadas.

Os algoritmos de aprendizado de máquina podem ser divididos em três categorias, dependendo da maneira com que utilizam a informação dos dados de entrada e seus respectivos rótulos (Tan, Steinbach e Kumar, 2006).

- **Aprendizado supervisionado**: os algoritmos desta categoria aprendem uma função a partir dos dados de entrada e dos seus rótulos associados. Baseados em um determinado número de exemplos, que são usados no treinamento, os algoritmos devem ser capazes de predizer os rótulos de dados ainda não vistos.

Para um bom funcionamento, os algoritmos devem ter capacidade de generalização, para que possam prever corretamente a saída de dados ainda não vistos. Problemas de classificação (classes com valores discretos) e regressão (classes com valores contínuos) fazem parte desta categoria.

- **Aprendizado não-supervisionado**: nesta categoria as amostras de dados não são rotuladas e os algoritmos precisam encontrar padrões nos dados de entrada. O agrupamento de dados e as regras de fazem parte do aprendizado não supervisionado. Em agrupamento, os dados de entrada são divididos em grupos baseados em sua similaridade; nas regras de associação são encontrados elementos que co-ocorrem em comum dentro de registros da base de dados;
- **Aprendizado semi-supervisionado**: utilizam um número pequeno de dados rotulados e uma grande quantidade de dados não rotulados para o treinamento, buscando assim a construção de classificadores que exigem menor esforço humano para a rotulação dos dados. O auto-treinamento (Zhu, 2005; Chapelle et *al*., 2006) e o co-treinamento (Blum e Mitchell, 1998) são métodos de aprendizado semi-supervisionado.

3.2 Detecção de *outliers*

Muitas definições têm sido propostas para *outliers*, Grubbs (1969) define uma observação *outlier* como sendo aquela que parece desviar-se acentuadamente dos outros membros da amostra em que ela ocorre. Outra definição apresentada por Barnett e Lewis (1994) diz que uma observação *outlier* (ou um subconjunto de observações) é aquela que parece ser inconsistente com o resto do conjunto de dados. Para Tan, Steinbach e Kumar, (2006) *outliers* são objetos de dados que tem características diferentes da maioria dos outros objetos num conjunto de dados.

A detecção de *outlier* pode ser utilizada em diversos casos, auxiliando na identificação de situações atípicas, como por exemplo, monitorar o uso de cartão de crédito e de celular, para detectar uma mudança brusca no padrão de uso que pode indicar uso fraudulento.

É possível também detectar falhas em uma linha de produção monitorando constantemente características específicas dos produtos e comparando os dados em tempo real de cada produto normal com aqueles em busca de falhas.

A detecção de *outlier* é uma tarefa importante em ambientes de segurança, podendo indicar condições anormais de funcionamento a partir das quais pode resultar uma degradação significativa no funcionamento. Um *outlier* pode denotar um objeto anômalo em uma imagem

ou um intruso dentro de um sistema com intenções maliciosas, como por exemplo, em uma rede de computador.

Para o processamento de aplicações, como processamento de pedido de empréstimo ou pagamento de prestações da segurança social, um sistema de detecção de *outlier* pode detectar eventuais anomalias no aplicativo antes da aprovação ou pagamento.

Comerciantes podem usar métodos de detecção de *outlier* para monitorar ações individuais ou mercados e detectar tendências inovadoras que podem indicar oportunidades de compra e venda.

Alguns fatores que causam o aparecimento de *outliers* podem ser erros humanos, de instrumentos, desvios em populações, comportamento fraudulento, mudanças ou falhas no comportamento de sistemas. Porém há dados que apresentam naturalmente pontos *outliers*.

Hodge e Austin (2004) descrevem três abordagens principais para o problema de detecção de *outlier*:

1. A primeira abordagem determina os *outliers* sem nenhum conhecimento prévio dos dados. É semelhante ao agrupamento não supervisionado. Considera os dados como uma distribuição estática, identifica os pontos mais remotos e os marca como potenciais *outliers*. Esta abordagem assume que os erros ou falhas são separados dos dados "normais" e assim, aparecem como *outlier*. Na Figura 7, os pontos o_1, o_2 e O_3 são pontos remotos separados dos grupos principais, N_1 e N_2, e seriam marcados como possíveis *outliers*. Há duas sub-técnicas normalmente empregadas: *diagnóstico* e *acomodação*. Uma abordagem por diagnóstico destaca os potenciais pontos *outliers*, que uma vez detectados, o sistema pode removê-los de processamentos futuros. Muitas abordagens podam iterativamente os *outliers* e ajustam o modelo para os dados restantes, até que não sejam mais detectados *outlier*s. Uma metodologia alternativa é a acomodação, a qual incorpora os *outliers* no modelo e emprega um método de classificação robusto, que pode suportar *outliers* nos dados.

2. A segunda abordagem é análoga à classificação supervisionada e exige que os dados estejam pré-rotulados como normais ou anormais. Na Figura 7, existiriam três classes de dados com *outliers* pré-rótulados em áreas isoladas, o_1, o_2 e O_3. Os pontos normais poderiam ser classificados em duas classes, N_1 e N_2. Se o exemplar fica em uma região de normalidade é classificado como normal, caso contrário é marcado como *outlier*. Este tipo de algoritmo baseado em classificação requer uma ampla cobertura de dados normais e anormais, para permitir a generalização do classificador.

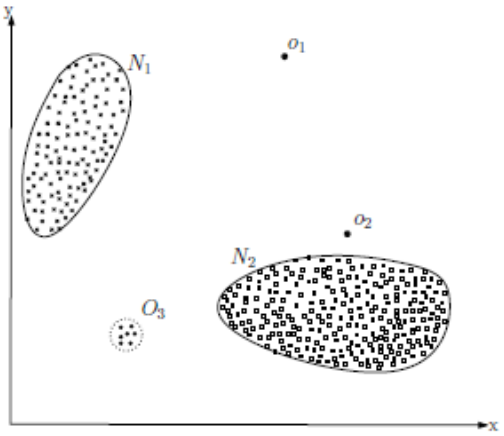


Figura 7: Exemplo simples de *outliers* em um conjunto de dados 2-D. Adaptado de (Chandola, Banerjee e Kumar, 2007).

3. A terceira abordagem é análoga ao reconhecimento semi-supervisionado no sentido de que a classe "normal" é ensinada e ele aprende a reconhecer anormalidade. Dessa maneira, a abordagem necessita apenas de dados pré-classificados como "normais". Isso é uma vantagem, uma vez que dados anormais normalmente são difíceis ou caros de obter. Neste método conforme chegam novos dados, se estes ficam fora dos limites da normalidade serão classificados como fraudes. Porém se a normalidade se afasta do modelo inicial, o método precisa reaprender os dados.

Chandola, Banerjee e Kumar (2007) descrevem as seguintes técnicas utilizadas para a detecção de *outliers:*

Técnicas baseada em classificação

Técnicas de detecção de *outlier* baseadas em classificação utilizam "normal" e "*outlier*" como rótulo das classes. Essas técnicas pertencem a segunda abordagem, ou seja, detecção de *outlier* supervisionada. Chandola, Banerjee e Kumar (2007) incluem neste grupo técnicas baseadas em redes neurais, redes bayesianas, árvores de decisão, modelos de regressão e análise de associação.

Técnicas baseada em agrupamento

Técnicas baseadas em agrupamento assumem que objetos de dados normais pertencem a grupos grandes e densos, enquanto objetos de dados *outliers* não pertencem a nenhum grupo ou formam grupos muito pequenos. Estas técnicas podem ser divididas em semi-supervisionadas e não-supervisionadas. As técnicas semi-supervisionadas normalmente usam dados normais para gerar grupos que representam o comportamento normal dos dados, um objeto novo de teste é alocado a um grupo, se não estiver próximo de nenhum é categorizado como *outlier*. Chandola, Banerjee e Kumar (2007) descrevem que técnicas como *Self-*

Organizing Maps (SOM), *K-means Clustering*, *Expectation Maximization* (EM) e *bootstrapping* já foram utilizadas. Técnicas não-supervisionadas usam um algoritmo conhecido de agrupamento para agrupar os dados e analisam cada instância com relação aos grupos formados.

Técnicas baseada em *K*-vizinho mais próximo

Tais técnicas assumem que os objetos de dados normais possuem vários vizinhos próximo deles, enquanto *outliers* são localizados longe dos demais pontos. Estas técnicas operam em dois passos: no primeiro passo uma vizinhança para cada dado é computada, usando uma medida de distância ou de similaridade entre dois objetos de dados. No segundo passo, a vizinhança é analisada para determinar se o objeto de dado é normal ou *outlier*.

Técnicas baseada em estatística

Estas técnicas desenvolvem modelos estatísticos (normalmente para um comportamento normal) a partir dos dados e então aplicam um teste de inferência estatística para determinar se um objeto de dado pertence ou não ao modelo. Objetos que possuem baixa probabilidade de pertencer ao modelo estatístico são declarados *outliers*.

Técnicas baseadas em teoria da informação

Estas técnicas analisam o significado das informações dos dados usando medidas como entropia, entropia condicional, ganho de informação entre outras. Assume-se que dados normais são regulares a certas medidas de teoria da informação e *outliers* alteram o significado da informação devido a sua natureza. Estas técnicas detectam objetos que induzem irregularidade nos dados, onde a regularidade é medida por uma medida particular de teoria da informação.

Técnicas baseadas em decomposição espectral

Estas técnicas tentam encontrar uma aproximação para os dados utilizando uma combinação de atributos que capturam a dimensão e variabilidade dos dados. Muitas técnicas utilizam *Principal Component Analysis* (PCA) para aproximação dos dados.

Técnicas baseadas na visualização

Estas técnicas tentam mapear os dados em um espaço de coordenadas que facilita a identificação de *outliers* visualmente. Um problema com estas técnicas é que são computacionalmente caras e difíceis de estender para altas dimensões.

3.3 Classificação de dados

Os dados de entrada para a classificação são uma coleção de registros. Cada registro é caracterizado por uma tupla (x, y), onde x é o conjunto de atributos e y é o rótulo das classes. A classificação pode ser definida como a tarefa de aprender uma função que seja capaz de predizer o rótulo de saída para uma nova entrada de dados do mesmo contexto, após ter avaliado certo número de exemplos de treinamento.

Exemplos de classificadores são árvores de decisão, redes neurais, aprendizado Bayesiano, entre outros. Cada técnica emprega um algoritmo de aprendizado para identificar um modelo que melhor representa a relação entre o conjunto de atributos e o rótulo de classe dos dados de entrada. O modelo gerado por um algoritmo de aprendizado deve representar bem os dados de entrada e predizer corretamente o rótulo da classe dos registros que nunca foram vistos antes.

Uma abordagem geral para resolver um problema de classificação é providenciar inicialmente um *conjunto de treinamento*, o qual consiste de registros cujo rótulo da classe é conhecido. Este conjunto de treinamento é então utilizado para construir o modelo de classificação, que em seguida deve ser aplicado a um *conjunto de teste*, que consiste de registros cujo rótulo de classe também é conhecido.

A avaliação do desempenho do classificador é baseada na contagem dos registros de teste corretamente e incorretamente preditos pelo modelo. Essa contagem pode ser marcada em uma tabela conhecida por *matriz de confusão*. Um exemplo retirado de Tan, Steinbach e Kumar (2006) pode ser visto na Tabela 1, a qual representa um problema de classificação binário. Cada entrada f_{ij} na tabela denota o número de registros da classe i classificado como sendo da classe j. O número total de classificação correta feita pelo modelo é ($f_{11} + f_{00}$) e o número total de classificação incorreta é ($f_{10} + f_{01}$).

Tabela 1: Matriz de confusão para um problema de 2 classes (Tan, Steinbach e Kumar, 2006).

		Classe predita	
		Classe = 1	*Classe = 0*
Classe atual	*Classe = 1*	f_{11}	f_{10}
	Classe = 0	f_{01}	f_{00}

A informação da matriz de confusão pode ser resumida em um único número através de uma medida de desempenho, como a A curácia, Equação (36), facilitando com isso a comparação de diversos modelos. Outros métodos para avaliação de desempenho dos classificadores, como *holdout*, *cross-validation* e *random subsampling* podem ser consultados em Tan, Steinbach e Kumar (2006).

$$Acurácia = \frac{número\ de\ predições\ corretas}{número\ de\ predições\ incorretas} = \frac{f_{11}+f_{00}}{f_{11}+f_{10}+f_{01}+f_{00}} \quad (36)$$

No que segue, é apresentada uma revisão sucinta de algumas técnicas consideradas básicas para a classificação de dados.

3.3.1 Árvores de decisão

Uma árvore de decisão é uma estrutura hierárquica que consiste de nós e arestas direcionadas. A árvore tem três tipos de nós, um *nó raiz* que não tem nenhuma aresta de entrada e possui uma ou mais arestas de saída. *Nós internos*, que possuem apenas uma aresta de entrada e duas ou mais arestas de saída. *Nós terminais* ou *nós folhas*, que possuem uma aresta de entrada e nenhuma aresta de saída.

Para cada nó terminal é atribuído um rótulo de classe. O nó raiz e os nós internos contêm testes de atributos que separam registros com características diferentes. Um exemplo de árvore de decisão pode ser visto na Figura 8.

A classificação de um registro de teste é simples, uma vez que a árvore de decisão foi construída. A partir do nó raiz, aplica-se a condição de teste para o registro e com base no resultado do teste acompanha-se o ramo adequado. Isso levará para outro nó interno, de forma que uma nova condição de teste seja aplicada, ou para um nó folha de forma que um rótulo de classe associado ao nó folha seja então atribuído ao registro.

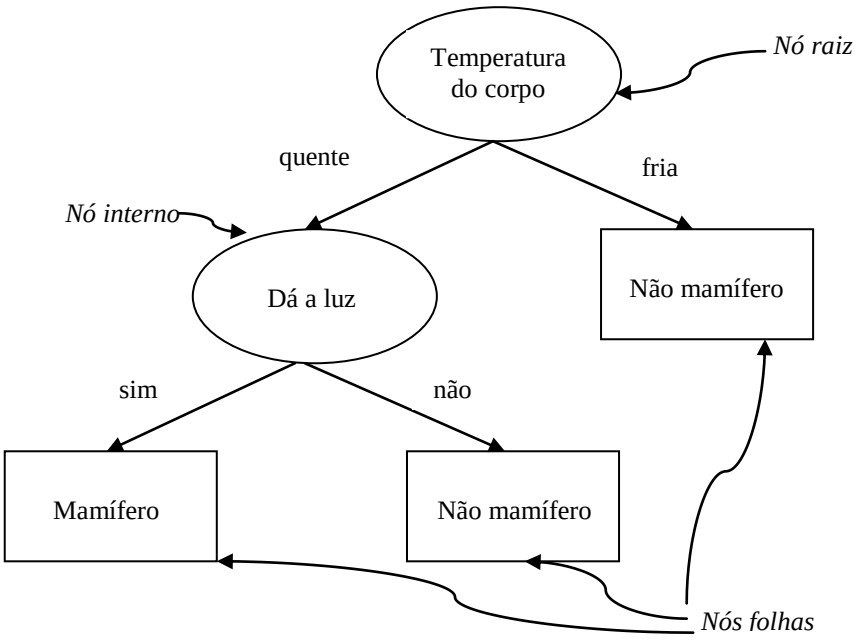


Figura 8: Uma árvore de decisão para um problema de classificação de mamíferos. Baseada em Tan, Steinbach e Kumar (2006).

Existem muitas árvores de decisão que podem ser construídas a partir de um dado conjunto de atributos. Algumas árvores são mais precisas do que outras, porém encontrar a árvore ótima é computacionalmente inviável devido ao tamanho exponencial do espaço de busca. No entanto, algoritmos eficientes têm sido desenvolvidos para induzir uma árvore de decisão razoavelmente precisa em um tempo aceitável. Esses algoritmos geralmente empregam uma estratégia gulosa que constroem uma árvore de decisão fazendo uma série de decisões ótimas locais sobre qual atributo usar para particionar os dados. Um desses algoritmos é o *Algoritmo de Hunt*, que é base para muitos outros algoritmos de árvores de decisão, como o ID3, C4.5 e CART (Tan, Steinbach e Kumar, 2006).

3.3.2 Redes neurais artificiais

As redes neurais artificiais (RNAs) são sistemas paralelos distribuídos formados por unidades de processamento, denominadas neurônios artificiais, sendo estas responsáveis pelo cálculo de certas funções matemáticas. Tais unidades são dispostas em uma ou mais camadas e interligadas por muitas conexões, sendo que estas podem estar associadas a algum peso, os quais armazenam o conhecimento adquirido e servem para ponderar as entradas dos neurônios (Braga et al., 2007).

As RNAs tentam reproduzir as funções das redes biológicas, buscando refletir sua dinâmica e comportamento. Como características comuns pode-se citar que os sistemas são baseados em unidades de computação paralela e distribuída que se comunicam por meio de conexões sinápticas, possuem detectores de características, redundância e modularização das conexões.

O primeiro modelo artificial de um neurônio biológico foi o trabalho de McCulloch e Pitts (1943), que tentava representar e modelar eventos no sistema nervoso. Porém somente depois de alguns anos da publicação do trabalho de McCulloch e Pitts que as redes neurais tornaram-se objeto de estudos.

3.3.2.1 Perceptron

O modelo conhecido como perceptron foi introduzido por Rosenblatt (1958). Uma ilustração é apresentada na Figura 9, o qual consiste de nós de entrada, que são usados para representar os atributos de entrada, e nós de saída, que são usados para representar o modelo de saída. Os nós de entrada são ligados por um *link* ponderado ao nó de saída. Esta ponderação é usada para simular a força das conexões e o treinamento do perceptron consiste

em ajustar os pesos até que suas saídas tornem-se consistentes com as saídas verdadeiras do conjunto de treinamento.

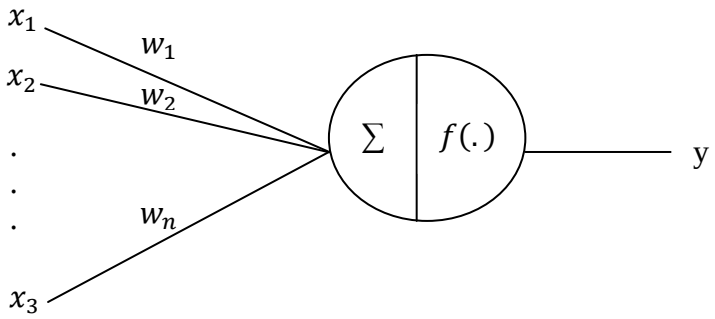


Figura 9: Neurônio perceptron. A saída *y* é obtida através da aplicação da função de ativação $f(.)$ sobre a soma ponderada das entradas $y = \sum_{i=1}^{n} x_i w_i$.

O perceptron calcula sua saída, *y*, fazendo uma soma ponderada sobre suas entradas, e então examinando o sinal do resultado:

$$y = \begin{cases} 1, se \ \sum_{i=1}^{n} x_i w_i > 0 \\ -1, se \ \sum_{i=1}^{n} x_i w_i < 0 \end{cases} \quad (37)$$

O sinal da função age como uma função de ativação para o neurônio de saída, onde a saída vale 1 se seu argumento é positivo e -1 se seu argumento é negativo.

Destaca-se que o perceptron garante a convergência para uma solução ótima em problemas de classificação linearmente separáveis. Se o problema não é linearmente separável o algoritmo falha na convergência.

3.3.2.2 Redes neurais artificiais multicamadas

Uma rede neural multicamadas tem uma estrutura mais complexa que o perceptron, pode conter camadas intermediárias entre as camadas de entrada e saída, as quais são chamadas camadas escondidas (Tan, Steinbach e Kumar, 2006). Além disso, a rede usa outras funções de ativação como linear, sigmoidal logística ou tangente hiperbólica, as quais permitem aos nós escondidos e de saída produzirem valores de saída não linear nos parâmetros de saída. A Figura 10 ilustra um modelo de rede neural multicamada.

Uma técnica conhecida por *backpropagation* usa o gradiente descendente para estimar o erro da camada de saída, assim o erro de saída da rede é calculado e retroalimentado para as camadas intermediárias, possibilitando o ajuste dos pesos proporcionalmente aos valores das conexões entre camadas.

Há duas fases em cada iteração do algoritmo: *forward* e *backward* (Tan, Steinbach e Kumar, 2006). Durante a fase *forward*, os pesos obtidos nas iterações anteriores são usados para calcular o valor de saída de cada neurônio na rede. A saída dos neurônios no nível *k* são calculados antes de computar a saída dos neurônios no nível *k* + 1. Durante a fase *backward*, a fórmula de atualização dos pesos é aplicada na direção oposta. Ou seja, os pesos no nível *k* + 1 são atualizados antes dos pesos no nível *k*. Esta abordagem de *backpropagation* permite usar os erros dos neurônios na camada *k* + 1 para estimar os erros dos neurônios na camada *k*.

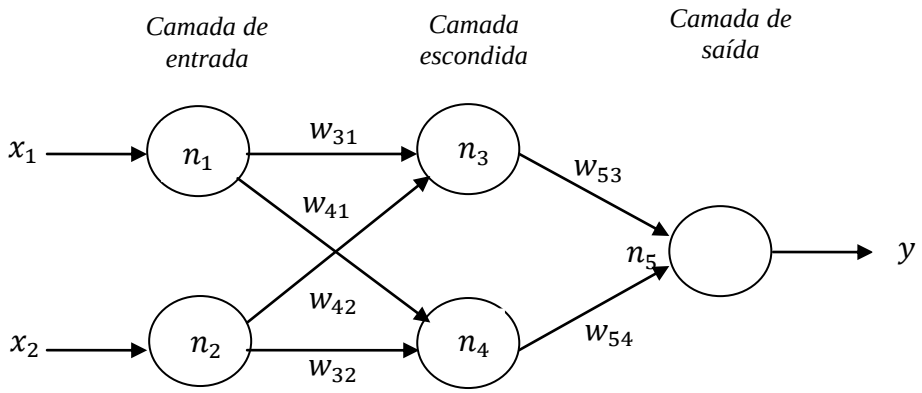


Figura 10. Rede neural com duas camadas.

Algumas questões que devem ser consideradas na modelagem de uma rede neural tratam da determinação do número de nós nas camadas de entrada e de saída; da topologia certa (número de camadas e nós escondidos, arquitetura *feed-forward* ou recorrente); inicialização dos pesos e seleção do conjunto de treino.

As redes neurais com duas camadas são aproximadores universais, podendo ser usadas para aproximar qualquer função. São pouco sensíveis a presença de ruídos no conjunto de treinamento e podem lidar com características redundantes, pois os pesos são automaticamente aprendidos durante a etapa de treinamento. Porém, treinar uma rede neural pode consumir muito tempo, principalmente se o número de camadas escondidas for grande (Tan, Steinbach e Kumar, 2006).

3.3.3 *K*-vizinhos mais próximos

Um classificador *k*-vizinhos mais próximos (*k-nearest neighbors*) representa cada exemplo como um ponto de dado em um espaço *n*-dimensional, onde *n* é o número de atributos. Tomando um exemplo de teste *x*, calcula-se sua proximidade aos demais pontos de dados, através de uma medida de (di)similaridade, como: Euclidiana, Pearson, cosseno, entre outras (Tan, Steinbach e Kumar, 2006). Os *k*-vizinhos mais próximos de um dado exemplo *x* são os *k* pontos de dados mais próximos de *x*. A Figura 11 ilustra 1, 2 e 3-vizinhos mais

próximos de um ponto de dado localizado no centro de cada círculo. O dado é classificado com base na classe de seus vizinhos, sendo que será atribuída a ele a classe majoritária dos vizinhos mais próximos. Em caso de empate será escolhida aleatoriamente uma delas para classificar o dado.

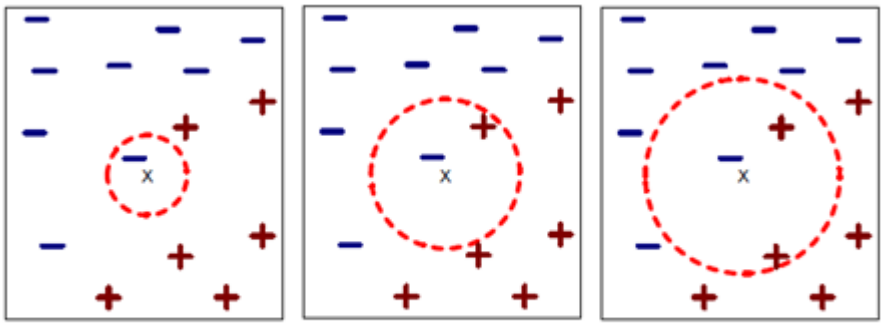


Figura 11. Exemplo de 1, 2 e 3-vizinhos mais próximos do dado *x*, adaptada de Tan, Steinbach e Kumar, (2006).

Uma questão que deve ser levada em consideração é a escolha do *k*. Se ele for muito pequeno, o classificador pode ter problemas devido a ruídos nos dados de treinamento. Se *k* for muito grande, o classificador também pode classificar incorretamente, pois os vizinhos que estão sendo considerados estão localizados longe da vizinhança do objeto de dado.

Esse tipo de classificador não requer a construção de um modelo, porém o custo computacional pode ser elevado uma vez que é necessário calcular a distância entre cada exemplo de teste e todos os demais exemplos de treinamento. As predições são feitas com base na informação local, por isso classificadores com valores pequenos de *k* são suscetíveis a ruído. Um ponto positivo é que ele pode produzir fronteiras de decisão de forma arbitrária fornecendo um modelo mais flexível em relação a uma árvore de decisão baseada em regras, que sempre adota fronteiras de decisão retilíneas.

3.3.4 Classificador Naïve Bayes

Existem diversas aplicações nas quais a relação entre os atributos e a classe é não determinística, ou seja, o rótulo da classe de um atributo de teste não pode ser predito com certeza. Isto pode acontecer devido a presença de ruídos ou de fatores de incertezas que afetam a classificação e não são incluídos na análise (Tan, Steinbach e Kumar, 2006). Uma abordagem para modelar relações probabilísticas entre os atributos e a classe são os classificadores bayesianos, que são baseados no teorema de Bayes, descrito a seguir.

Sejam X e Y variáveis aleatórias. A probabilidade conjunta se refere a probabilidade de dois eventos X e Y ocorrerem. Temos então:

- $P(X)$: a probabilidade do evento X ocorrer.
- $P(Y)$: a probabilidade do evento Y ocorrer.
- $P(X\ \&\ Y)$ ou $P(X, Y)$: a probabilidade de X e Y ocorrerem.
- $P(X\ \&\ Y) = P(X) * P(Y)$ se X e Y forem eventos independentes, ou seja, a ocorrência de um não afeta a probabilidade de ocorrência do outro.

Se X e Y não forem eventos independentes, tem-se $P(X\ \&\ Y) = P(X) * P(Y|X)$. Ou seja, $P(Y|X) = P(X\ \&\ Y)/P(X)$ expressa a probabilidade que Y ocorra dado que X ocorreu. Essa é a probabilidade condicional de Y dado X.

Como $P(X\ \&\ Y) = P(Y\ \&\ X)$ temos $P(Y|X) * P(X) = P(X|Y) * P(Y)$, e a partir disso o teorema de Bayes:

$$P(Y|X) = P(X|Y) * P(Y)/P(X) \tag{38}$$

Seja X o conjunto de atributos e Y a variável classe. Se a classe tem uma relação não determinística com os atributos, X e Y podem ser tratados como variáveis aleatórias e sua relação podem ser capturadas de forma probabilística usando-se $P(Y|X)$. Durante a fase de treinamento é necessário aprender $P(Y|X)$ para cada combinação de X e Y baseado na informação obtida nos dados de treinamento.

O classificador Naïve Bayes estima a probabilidade de classe condicional assumindo que os atributos são condicionalmente independentes. Com isso, é necessário calcular a probabilidade condicional de cada X_i, dado Y:

$$P(X|Y) = \frac{P(Y)\prod_{i=1}^{N} P(X_i|Y)}{P(X)} \tag{39}$$

3.3.5 Classificação relacional

Os conjuntos de dados podem ser representados por diversas formas, influenciando com isso as técnicas de aprendizado de máquina que são aplicadas. As principais técnicas são baseadas na representação atributo-valor, porém existem técnicas que trabalham com conjuntos de dados relacionais, e técnicas que utilizam as duas representações dos dados.

Geralmente, as técnicas de classificação relacional baseada em grafos utilizam técnicas de inferência coletiva para induzir valores dos rótulos destes exemplos, estimando a classe de cada exemplo não rotulado ou sua distribuição de probabilidade. A distribuição de probabilidade de um exemplo é a probabilidade deste pertencer a cada classe.

A seguir são apresentados três métodos de inferência coletiva, a amostragem de Gibbs, ou *Gibbs Sampling* (Geman e Geman, 1984), a Relaxação de Rótulos, ou *Relaxation Labeling* (Chakrabarti et al., 1998) e a Classificação Iterativa, ou *Iterative Classification* (Lu e Getoor, 2003).

3.3.5.1 Inferência coletiva

Inferência coletiva significa inferir simultaneamente valores inter-relacionados, pode ser aplicada a dados em redes para estimar a classe de cada exemplo não rotulado ou sua distribuição de probabilidade. Uma vantagem de se estimar as classes por inferência coletiva é que não é necessário descartar os exemplos não rotulados durante a classificação, obtendo uma distribuição de probabilidade ou classe estimada para todos os exemplos (Macskassy e Provost, 2007).

As técnicas de inferência coletiva iniciam o procedimento com a utilização de um modelo de classificação local para estimar uma distribuição de probabilidade inicial para cada exemplo não rotulado, seguido de um processo iterativo, no qual se utiliza um modelo de classificação relacional para atualizar a distribuição de probabilidade de cada exemplo, e o processo é interrompido, em geral, quando os valores se estabilizam.

A seguir são apresentados três métodos de inferência coletiva.

Amostragem de Gibbs

Geman e Geman (1984) publicaram inicialmente o algoritmo amostragem de Gibbs, Gelfand e Smith (1990) sugeriram posteriormente a aplicação do algoritmo para resolução de modelos estatísticos Baysianos.

A exigência básica para o algoritmo é que se tenha disponível todas as distribuições condicionais para os parâmetros do modelo. A partir de um vetor arbitrário de valores iniciais, uma sequência de amostras das distribuições dos parâmetros condicionais é gerada de forma iterativa convergindo para a distribuição de parâmetros comuns, independentemente da seleção de valores iniciais.

O algoritmo de amostragem de Gibbs foi implementado por Macskassy e Provost (2007) e possui cinco etapas principais:

1. Estimação de uma distribuição de probabilidade de cada exemplo não rotulado utilizando um modelo de classificação local M_L. A classe inicial de cada exemplo é, então, obtida por uma amostra considerando sua distribuição de probabilidade. A definição de classe inicial do exemplo segue um esquema de roleta, priorizando as classes com maiores valores de probabilidade.

2. Geração de uma ordenação O desses exemplos não rotulados.

3. Utilização de um modelo de classificação relacional M_R para cada elemento x_i de O para estimar a classe de x_i, também por amostragem a partir da distribuição de probabilidade gerada pelo classificador M_R, sendo que para obter a nova classe do exemplo x_i são utilizadas sempre as classes mais recentemente obtidas, incluindo as "novas" classes de x_1, ..., x_{i-1}.

4. Repetição do Passo 3 até que a distribuição de probabilidade de cada exemplo para a ordenação aleatória O se estabilize.

5. Repetição do processo iterativo (Passos 2, 3, 4) até que a ordenação aleatória O dada em cada repetição não influencie no resultado, obtendo a quantidade de vezes que cada classe foi definida para cada exemplo ao final de cada iteração, normalizando essa contagem para se obter uma distribuição de probabilidade final de cada exemplo.

Relaxação de rótulos

O método de relaxação de rótulos foi proposto por Chakrabarti et al. (1998) e a cada iteração t a distribuição de probabilidade de cada exemplo não rotulado é alterada, considerando a distribuição $t - 1$ dos adjacentes desse exemplo. As principais etapas do algoritmo são:

1. Utilização de um modelo de classificação M_L para obter-se a distribuição de probabilidade inicial $d_0(x_i)$ para cada exemplo x_i não rotulado.

2. Aplicação do modelo de classificação relacional M_R em cada elemento x_i do conjunto de exemplos não rotulados para obter sua nova distribuição de probabilidade $d_t(x_i)$, utilizando para isso as distribuições de probabilidade d_{t-1} dos vizinhos de x_i na rede.

3. Repetição do Passo 2 até que todos os valores estabilizem-se.

Como em certos casos a relaxação de rótulos fica oscilando entre dois ou mais estados não convergindo para uma situação estável, Macskassy e Provost (2007) adaptaram o método fazendo com que em cada iteração t a estimativa obtida para o exemplo x_i na iteração $t - 1$ tivesse mais peso e a nova distribuição de probabilidade obtida na iteração t tivesse menos peso.

Para isso, fez-se necessário uma alteração na forma de atribuição de uma nova distribuição de probabilidade a um exemplo. Para se estimar a nova distribuição de probabilidade da iteração $t + 1$ (definida como $nd_{t+1}(x_i)$) com peso, é necessário considerar,

além da distribuição $d_{t+1}(x_i)$ obtida pelas distribuições d_t dos vizinhos, também a distribuição $nd_t(x_i)$ da iteração t:

$$nd_{t+1}(x_i) = \beta_{t+1} \cdot d_{t+1}(x_i) + (1 - \beta_{t+1}) \cdot nd_t(x_i) \quad (40)$$

na qual β_0 é iniciado entre 0 e 1 e $\beta_{t+1} = \beta_t \cdot \alpha$ com α sendo uma constante de decaimento.

Classificação iterativa

A classificação iterativa foi proposta por Lu e Getoor (2003) e não gera probabilidade, mas estima uma determinada classe para todos os exemplos não rotulados. O método apresenta as seguintes etapas:

1. Utilização de um modelo de classificação local M_L para obtenção de uma classe para cada exemplo não rotulado.
2. Geração de uma ordenação O dos exemplos não rotulados pela quantidade de diferentes classes existentes em seus adjacentes, já que a confiança na classificação dos exemplos com menor diversidade de classes nos adjacentes é maior. Para cada exemplo não rotulado em O aplica-se o modelo de classificação relacional M_R, obtendo-se sua distribuição de probabilidade, e atribuindo-se a classe de maior probabilidade para o exemplo. Caso todos os adjacentes ainda não possuam classe atribuída então esse exemplo continua não rotulado.
3. Repetição do Passo 2 até que nenhum exemplo tenha sua classe alterada.

3.3.5.2 Classificadores relacionais

A seguir são apresentados dois classificadores que utilizam as informações individuais e relacionais dos exemplos, a classificação de Hipertexto, ou *Hypertext classification*, e a classificação baseada em links, ou *link-based classification*. Esses classificadores consideram duas formas de obtenção da distribuição de probabilidade de cada exemplo: a primeira é a distribuição de probabilidade conjunta, a qual é utilizada apenas para obter uma distribuição de probabilidade inicial para os exemplos não rotulados, obtida por métodos de inferência coletiva. A segunda é a distribuição de probabilidade marginal, a qual considera a vizinhança do exemplo na rede e é obtida por métodos de classificação relacional baseada em grafos.

Classificação de hipertexto

A classificação de hipertexto foi proposta por Chakrabarti et al. (1998) para classificação de dados relacionais com conteúdo textual como, por exemplo, páginas *web*. A probabilidade de um exemplo x_i pertencer a classe c é obtida a partir da distribuição de

probabilidade de um classificador local que utiliza o conteúdo textual dos documentos, e da distribuição de probabilidade de um classificador relacional baseado nos exemplos adjacentes a x_i, sendo representada pela Equação (41):

$$P(x_i = c|t_i, N_i) = P(x_i = c|t_i) \cdot P(x_i = c|N_i) \quad (41)$$

na qual t_i representa o conteúdo textual estruturado de x_i, e N_i os exemplos que estão em sua vizinhança na rede, a classe atribuída a x_i é a que maximiza essa probabilidade.

Para os exemplos não rotulados, os autores utilizam relaxação de rótulos para considerar a distribuição de probabilidades. Utilizando o conteúdo textual para o modelo de classificação relacional $P(x_i = c|t_i)$ e a vizinhança do vértice para o modelo de classificação relacional $P(x_i = c|N_i)$, atribuindo como classe aquela que maximizar a probabilidade.

Classificação baseada em *links*

A classificação baseada em *links* foi proposta por Lu e Getoor (2003), e utiliza um modelo de regressão logística para realizar a classificação. A regressão logística é utilizada em casos binários para se estimar a probabilidade de uma variável pertencer a cada uma das duas classes. Nos conjuntos de dados multi-classes os autores consideram, para cada classe, a probabilidade do dado pertencer e a probabilidade dele não pertencer à classe.

Para tal é utilizado um grafo direcionado, no qual são observadas as características individuais dos exemplos (atributos) e suas ligações na rede (vizinhança), e estas informações são armazenadas em dois vetores. O vetor baseado na vizinhança da rede pode ser representado pelo modelo *mode-link*, o qual atribui o valor 1 na posição da classe mais freqüente dos vizinhos e 0 no restante do vetor, o modelo *count-link*, o qual faz uma contagem do número de adjacentes de cada classe, e o modelo *binary-link*, o qual atribui 1 em cada posição do vetor caso o exemplo tenha algum adjacente da classe ou 0 caso contrário.

A partir destes vetores os autores propuseram uma classificação iterativa, que utiliza o modelo de inferência coletiva. A regressão logística é aplicada como modelo de classificação local e relacional, sendo que no modelo local utiliza-se apenas o vetor de atributos e no modelo relacional utiliza-se os dois vetores. A cada etapa do método iterativo computa-se a probabilidade de cada exemplo pertencer a cada classe, sendo este classificado como a classe de maior probabilidade. O processo é interrompido quando as classes atribuídas aos exemplos se estabilizam.

3.3.6 Classificação baseada em rede *K*-associados

Como os trabalhos que utilizam informações relacionais consideram conjuntos de dados que possuem inerentemente uma estrutura relacional, como as páginas *web*, citações de artigos científicos, entre outras, Lopes et al. (2009) propuseram uma alternativa para representar relacionalmente os conjuntos de dados proposicionais, por meio de grafos baseados na similaridade entre os objetos. A seguir é apresentado o modelo de redes denominado *K*-associados proposto, sua formação e classificador específico, que exploram características dessa rede.

3.3.6.1 Rede *K*-associados e medida de pureza

Em uma rede *K*-associados, entre os *K* vizinhos de um vértice dado, as conexões serão estabelecidas entre o vértice dado e aqueles pertencentes a uma mesma classe. Como este processo segue por todos os vértices da rede, em muitos casos um vértice v_i pode já estar conectado a um vértice v_j (supondo v_i e v_j pertencentes a uma mesma classe) quando o vértice v_j seleciona o vértice v_i para conectar, resultando em duas conexões diferentes entre os vértices v_i e v_j. Isto é justificado pelo fato de que se v_j está na *K*-vizinhança de v_i, então irá se estabelecer uma conexão. Dessa forma, quando v_i é encontrado na *K*-vizinhança de v_j uma conexão é estabelecida não importando se eles já estão conectados.

De uma maneira formal, a rede *K*-associada resultante *A* = (*V*, *E*) consiste de um conjunto de vértices rotulados *V* e um conjunto de arestas *E* entre eles, onde uma aresta e_{ij} conecta o vértice v_i com o vértice v_j se a classe (v_i) = classe (v_j) e v_j pertence aos *K*-vizinhos mais próximos de v_i. Onde a classe (v_i) é o rótulo ou a classe de v_i.

Seja $k_{nn}(v_i)$ o conjunto dos *K*-vizinhos mais próximos do vértice v_i por uma medida de similaridade dada, observe que vértices em $k_{nn}(v_i)$ podem ter diferentes rótulos de classes. A vizinhança N_{K_i} para um vértice v_i é definida como seus *K*-vizinhos conectados como segue: $N_{K_i} = \{v_j \mid e_{ij} \in E \; e \; v_j \in classe\,(v_i) \; e \; v_j \in k_{nn}(v_i)\}$. O grau g_i é definido como o número de arestas para seus vizinhos N_{K_i} e isto é no máximo $2K_i$ (*K* arestas de v_i para seus vizinhos e *K* de seus vizinhos para v_i).

O método de geração da rede *K*-associada melhora a representação dos dados e permite alguns cálculos interessantes como a medida de *pureza*. Essa medida quantifica quão conectados estão os vértices da mesma classe.

A Figura 12 ilustra uma representação bi-dimensional de um conjunto de dados contendo 10 exemplos com rótulo preto e 5 com rótulo branco, e suas correspondentes redes 1, 3 e 5-associados.

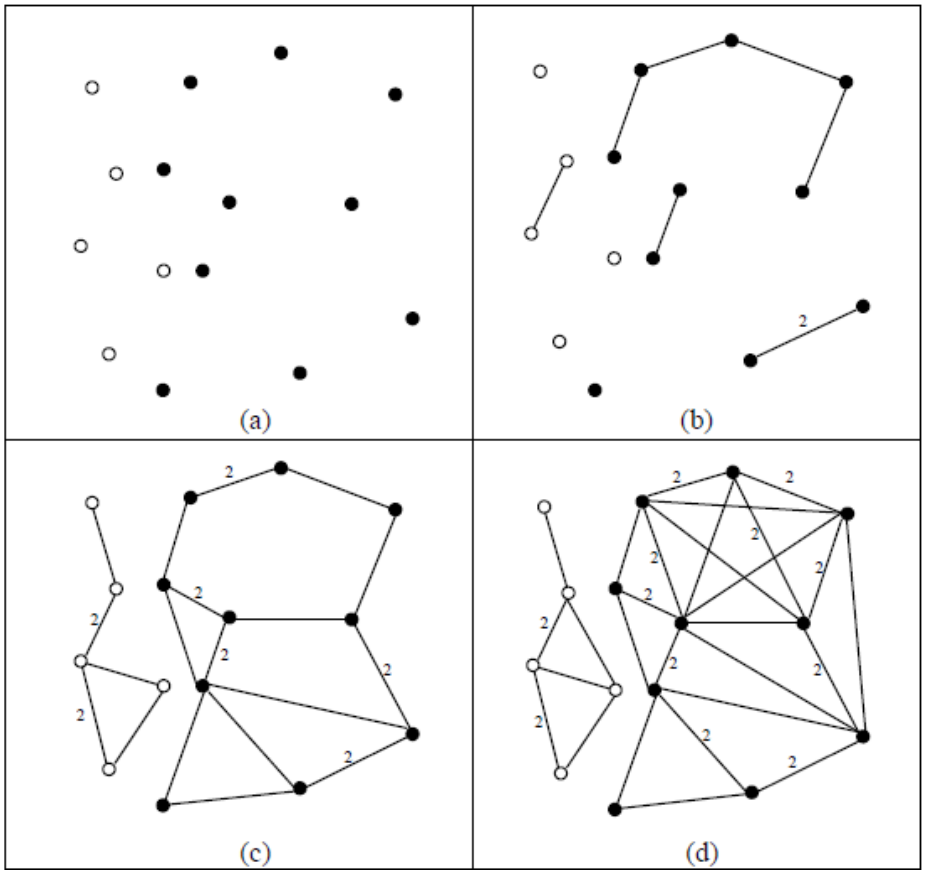


Figura 12: Redes *K*-associados: (a) Representação em 2D da base de dados. (b) (c) e (d) são as redes 1, 3 e 5-associados. Em algumas redes as arestas entre os vértices podem representar duas conexões. Obtida em (Lopes et al., 2009).

Seja g_i o grau do vértice v_i, *N* o número de dados no conjunto de treinamento (número de vértices), *K* um parâmetro para controlar o número de vizinhos usados na construção da rede. A fração $g_i/2K$ corresponde a conexão entre o vértice v_i e os vértices em seu próprio componente, note que cada classe pode possuir um ou mais componentes da rede e cada componente é um conjunto de vértices conectados. Essa proporção varia entre 0 e 1, inclusive. Em seguida, o total de conexões entre N_c vértices num componente *C* é dado pela Equação (42).

$$|E_c| = \frac{1}{2}\sum_{i=1}^{N_c} g_i = \frac{N_c}{2}\sum_{i=1}^{N_c}\frac{g_i}{N_c} = \frac{N_c}{2}\langle G_c\rangle \tag{42}$$

Onde $\langle G_c\rangle$ corresponde ao grau médio do componente *C*. O número máximo de arestas entre N_c vértices é KN_c desde que $K < N_c$. Com isso, a probabilidade de arestas entre vértices no mesmo componente *C* (componentes inter conectados) é dada pela Equação (33).

$$P_i = \frac{\frac{N_C\langle G_C\rangle}{2}}{KN_c} = \frac{\langle G_C\rangle}{2K} \quad (43)$$

Na Equação (43) P_i = 1 quando há somente vértices com o mesmo rótulo na *K*-vizinhança de todo componente de v_i, ver Figura 13. Daí, $\langle G_c\rangle/2K$ pode ser visto como uma medida de *pureza* na região do componente *C*.

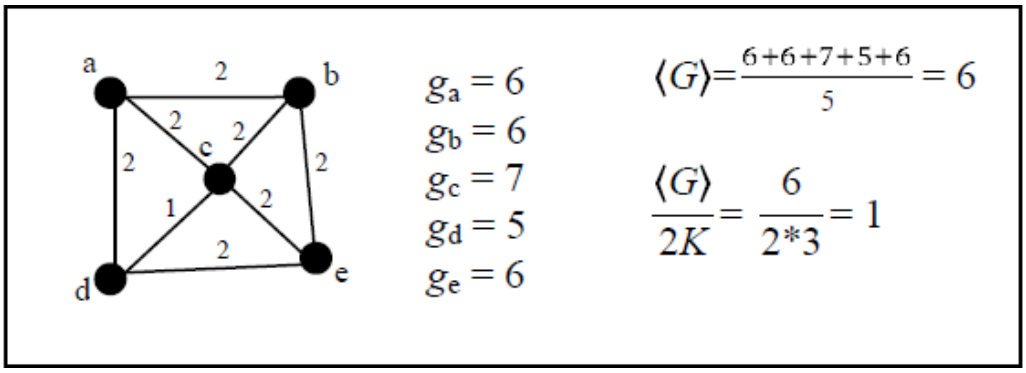


Figura 13: Um exemplo de um componente "puro" com 5 vértices e *K* = 3. Obtida em (Lopes et al., 2009).

No processo descrito até agora, cada *K* gera uma rede e certamente algumas redes terão componentes melhores do que outras, de acordo com a noção de pureza. Raramente uma rede obtida de um único *K* terá os melhores componentes entre todos os componentes em todas as *K* redes possíveis. A partir disso, o objetivo é obter uma rede com a melhor organização de todos os dados nos componentes independentemente de se ter um único *K*. Fazendo isto, a ideia é variar *K* mantendo os melhores componentes encontrados. Esse processo irá resultar em uma rede chamada rede ótima, com componentes formados por valores distintos de *K*.

A ideia da rede ótima baseada apenas na pureza dos componentes tem algumas desvantagens. Como a pureza não considera o tamanho do componente, ela tende a favorecer os menores. Uma forma intuitiva de superar este problema é multiplicar a pureza pelo número de vértices para um dado componente, como mostrado na Equação (44). Entretanto a medida *W* incorre em outro problema, i.e, muitos componentes com baixa pureza podem ter vantagens sobre uma minoria com alta pureza. Para resolver este problema, uma nova restrição que relata tamanho e pureza de forma indireta é adicionada na equação.

$$W_j = \frac{\sum_{i=1}^{N_j} g_i}{2K_j} \text{ e } \langle G_c\rangle > \text{K} \quad (44)$$

na qual W_j é a nova medida para o componente C_j, g_i é o grau do vértice v_i, N_j é o número de padrões no componente C_j, K_j é o número de vizinhos usado na construção do componente *j*.

Note que a medida de pureza ainda será usada, após a rede ótima ser obtida, no processo de classificação. A rede ótima é a estrutura final obtida por meio desse processo. Essa rede pode ser vista como o resultado do processo de aprendizagem supervisionado e será usada no processo de classificação como mostrado na próxima seção.

3.3.6.2 Classificador *K*-associado

No que segue, é apresentado o classificador *K*-associado não-paramétrico que usa a rede ótima *K*-associada como modelo dos dados de treinamento para classificar precisamente novos dados. Conforme mostrado anteriormente, essa estrutura armazena os melhores componentes dos dados encontrados através de uma larga gama de *K*. O componente de pureza pode ser visto como um primeiro dado representado no componente. Desde que cada componente contém vértices (instâncias) de apenas uma classe, podemos computar a probabilidade de uma nova instância pertencer a uma classe dada computando a probabilidade desta instância pertencer aos componentes da mesma classe. Antes de apresentar os detalhes deste classificador algumas notações devem ser introduzidas.

Tipicamente um padrão de treinamento x_i é representado por $\mathbf{x}_i = \left(x_{i1}, x_{i2}, \cdots, x_{ip}, w_i\right)$, tal que x_i representa o *i*-ésimo padrão de treinamento com w_i sendo sua classe associada, em um problema de *M*-classe $\Omega = \{w_1, w_2, \cdots, w_M\}$. Da mesma forma um novo padrão é definido como $\mathbf{y}_j = \left(y_{j1}, y_{j2}, \cdots, y_{jp}\right)$ exceto que agora a classe w_j associada com o novo padrão $\mathbf{y}_j$ deve ser estimada. Considerando também o grupo de componentes da rede ótima $\mathrm{C} = \{C_1, C_2, \cdots, C_R\}$ onde *R* é o número de componentes e $R \geq M$.

De acordo com a teoria de Bayes a probabilidade posterior de uma nova instância $\mathbf{y}_i$ pertencer ao componente C_j dado os vizinhos N_{K_i} de $\mathbf{y}_i$ que pertençam ao componente C_j é:

$$P\left(\mathbf{y} \in C_j | N_{K_i}\right) = \frac{P\left(N_{K_i} | C_j\right) P(C_j)}{P\left(N_{K_i}\right)} \tag{45}$$

É importante ter em mente que cada componente C_i vem de uma particular rede *K*-associada. Assim, a vizinhança N_{K_i} deve considerar este *K* particular.

Como a pureza marca individualmente quão puro é cada componente, a pureza normalizada age como uma primeira probabilidade:

$$P\left(C_j\right) = \frac{\langle G_{C_j} \rangle}{\sum_{i=1}^{R} \langle G_{C_j} \rangle} \tag{46}$$

A probabilidade de se ter N_{K_i} conexões entre os possíveis K_j, para o componente C_j, é:

$$P\left(N_{K_i}|C_j\right) = \frac{\#\left\{e_{i_p}|v_p \in C_j\right\}}{K_j} \tag{47}$$

onde **'#'** representa a cardinalidade do grupo. E finalmente a probabilidade de N_{K_i} condições é dada pela Equação (48).

$$P\left(N_{K_i}\right) = \sum_{i=1}^{M} P\left(N_{K_i}|C_i\right) P(C_i) \tag{48}$$

Como em muitos casos há mais componentes que classes, de acordo com o classificador ótimo de Bayes, é necessário resumir a probabilidade posterior que corresponde a uma classe comum.

Então, a probabilidade posterior da nova instância pertencer a uma classe dada é mostrada pela Equação (49).

$$P(\mathbf{y}|w_i) = \sum_{C_j=w_j} P\left(\mathbf{y} \in C_j|N_j\right) \tag{49}$$

Finalmente os maiores valores entre as probabilidades posteriormente encontradas refletem a mais provável classe alocada para a nova instância, de acordo com a Equação (50).

$$\varphi(\mathbf{y}) = \arg max \ \{P(\mathbf{y}|w_1), \cdots, P(\mathbf{y}|w_M)\} \tag{50}$$

onde $\varphi(\mathbf{y})$ fica para a classe atribuída para a instância **y**.

Capítulo 4

Detecção de *outliers* em rede complexas

Este capítulo apresenta um método para detecção de vértices *outliers* em redes complexas (Berton et al., 2010) baseado na distância da caminhada aleatória de uma partícula Browniana e no índice de dissimilaridade (Zhou, 2003a), (Zhou, 2003b), (Noh e Rieger, 2004), e permite que se leve em consideração informações locais e globais da rede, de forma que cada vértice tenha sua própria "visão" da rede. Aqueles vértices que tenham uma "visão" muito diferente dos demais são considerados *outliers*.

O método foi aplicado na identificação de *outliers* em redes artificiais e reais, os resultados obtidos mostram que os vértices *outliers* não são apenas aqueles mais distantes do centro da rede, ou simplesmente vértices com um grau muito baixo ou muito alto, mas também aqueles que possuem um comportamento global diferente da maioria. Desse modo, o método é capaz de identificar vários tipos de vértices *outliers* usando uma única medida.

No que segue o método é apresentado e alguns resultados obtidos com a aplicação do método em redes artificiais e reais são mostrados.

4.1 Medida de distância e índice de dissimilaridade

A caminhada aleatória em redes tem sido estudada por diversos pesquisadores (Woess, 2000; Zhou, 2003a; Zhou 2003b; Noh e Rieger, 2004). Em Zhou (2003b) é apresentado uma medida de distância baseada no movimento de uma partícula Browniana numa rede. O movimento de uma partícula Browniana em uma rede pode ser interpretado como uma caminhada aleatória, pois a cada passo a partícula opta arbitrariamente por mover-se para um de seus vizinhos, sem levar em consideração qualquer força. A medida de distância obtida é

usada por Zhou (2003a) para detecção de comunidades em redes. Aqui, nós usamos a medida de distância de uma caminhada aleatória para identificar vértices *outliers* em redes complexas.

Seja uma rede com *N* e *M* arestas, onde o conjunto de vértices é denotado por $V = \{1,2,\dots,N\}$ e as arestas entre pares de vértices são representadas por uma matriz de adjacência generalizada *A*. A probabilidade de uma partícula mover-se de um vértice *i* a um vértice *j* em uma iteração é dada por $P_{ij} = A_{ij} / \sum_{l=1}^{N} A_{il}$. A correspondente matriz *P* é chamada de matriz de transferência.

A distância de um vértice *i* até um vértice *j* corresponde ao número médio de passos necessários para uma partícula Browniana mover-se através da rede de um vértice *i* até um vértice *j*, pode ser calculada pela Equação (51).

$$d_{i,j} = P_{ij} + \sum_{m=1}^{\infty}(m+1) \sum_{k_1 \neq j;\dots;k_m \neq j} P_{ik_1} P_{k_1 k_2} \dots P_{k_m j} \tag{51}$$

na qual *m* é o número de passos entre *i* e *j*. Essa distância pode ser interpretada como o tempo médio necessário para uma partícula mover-se do vértice *i* até o vértice *j* através da rede. Para um tempo $t \gg M$ a probabilidade de uma partícula Browniana localizar-se em um vértice *k* é dada por $p(k) = \sum_l A_{kl} / \sum_{m,n} A_{mn}$, proporcional ao número de vizinho do vértice *k* . Uma vez que a matriz de transferência *P* satisfaz as características de uma matriz de transição irredutível de Markov (Anton e Rorres, 2005), onde $P_{i1} + P_{i2} + \cdots + P_{iN} = 1$, podemos aplicar o teorema de convergência de uma cadeia de Markov para um vetor ponto fixo, em que $X = P^n X$ quando $n \to \infty$, e obter a seguinte equação algébrica da distância:

$$[I - B(j)] \begin{pmatrix} d_{1j} \\ d_{1j} \\ \vdots \\ d_{1N} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \tag{52}$$

onde *I* é a matriz identidade $N\chi N$ e a matriz *B*(*j*) é igual a matriz de transferência *P*, exceto que $B_{lj} = 0$ para todo $l \in V$. Resolvendo a Equação (52), obtemos o vetor de distância $\{d_{1j}, d_{2j}, \dots, d_{Nj}\}^t$, com a distância de todos os vértices em *V* até um vértice *j*. Resolvendo então para todo $j \in V$, obtemos a matriz de distâncias Δ.

A distância $\Delta_{i,j}$ geralmente é diferente de $\Delta_{j,i}$, indicando que a perspectiva de dois vértices *i* e *j* apenas é a mesma se esses vértices possuem a mesma estrutura de conexão na rede.

4.2 O método de detecção de *outliers*

Nosso método é baseado na medida de distância de uma caminhada aleatória e no índice de dissimilaridade apresentado na Seção 4.1. O índice de dissimilaridade mede a diferença entre a perspectiva de dois vértices na rede e pode ser calculado por:

$$\Lambda\,(\mathrm{i},\mathrm{j}) = \frac{\sqrt{\sum_{k\neq i,j}^{N}(\Delta_{ik}-\Delta_{jk})^2}}{(N-2)} \tag{53}$$

Calculando $\Lambda\,(\mathrm{i},\mathrm{j})$ para todo par $\langle i,j\rangle \in V$ obtemos a matriz de dissimilaridade Λ, a qual é simétrica.

Considerando uma rede cuja dissimilaridade entre cada par de vértices é representada pela matriz Λ. Podemos identificar os vértices mais singulares em uma rede estabelecendo um *score* de *outliers* para cada vértice *i*. Escolhemos a soma da linha *i* da matriz Λ para prover este *score*, conforme a Equação (54):

$$\sigma(i) = \frac{1}{\sqrt{N}}\sum_{l=1}^{N}\Lambda(\mathrm{i},\mathrm{l}) \tag{54}$$

A partir do *score* $\sigma(i)$ ranqueamos o conjunto V em ordem descendente tal que os primeiros elementos do ranking têm o maior *score*.

4.3 Detecção de *outliers* em redes artificiais

Para um melhor entendimento do que representa a medida de *outlier* proposta, tomamos uma rede simples com 5 vértices, conforme representada na Figura 14(a). Ao aplicar o método sobre a rede, os vértices com maior *score* de *outliers* são o 1 e 5. Poderíamos inferir que o método identifica os vértices mais distantes do centro da rede como *outlier*. Seria um comportamento que se assemelha ao comportamento de muitos métodos clássicos de detecção de *outliers*, os quais identificam como *outliers* os vértices mais distantes dos centróides no espaço de características. O mapa de cores na Figura 14(b) mostra o índice de dissimilaridade para cada par de vértices e a Figura 14(c) mostra o *score* de *outlier* para cada vértice.

Tomando uma rede mais estruturada (com 9 vértices) representada na Figura 15(a), percebemos que, embora os vértices 1, 5, 7, e 9 sejam os mais distantes do centro da rede, o *score* obtido por estes através do método é superado pelo *score* obtido pelo vértice 3. Poderíamos inferir então, neste caso, que o método identifica os vértices com maior e com menor grau na rede como *outlie*r. O mapa de cores da Figura 15(b) mostra o índice de dissimilaridade entre cada par de vértices e a Figura 15(c) mostra o *score* de *outlier* para cada vértice.

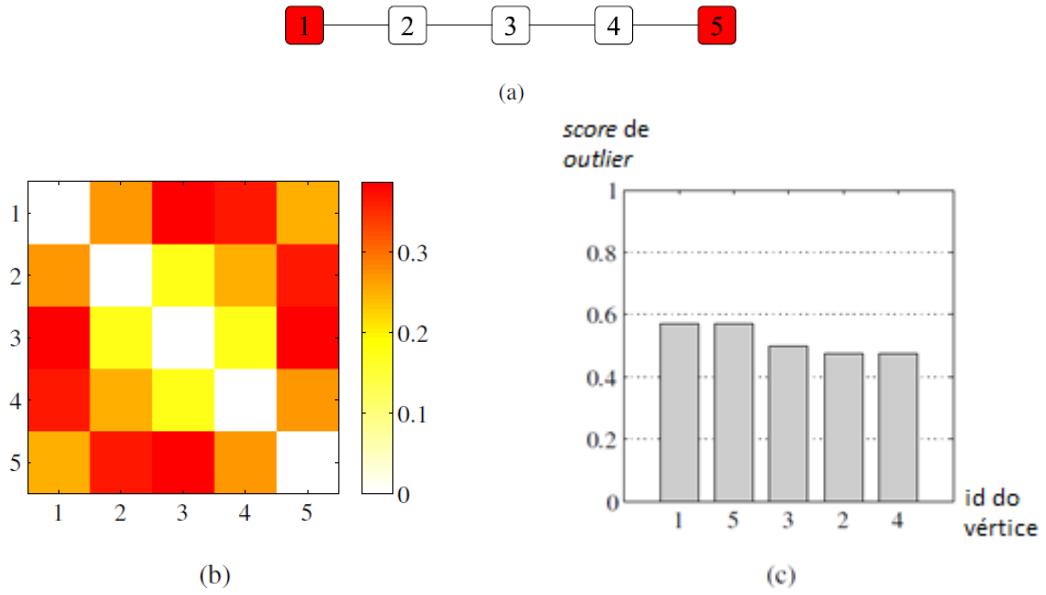


Figura 14: (a) Resultado do método de detecção de *outlier* aplicado em uma cadeia com 5 vértices. Os vértices vermelhos têm o maior *score* de *outlier*. (b) a matriz de dissimilaridade Λ representada por cores. Vermelho representa o valor mais alto e branco o mais baixo. (c) *Rank* de *outlier*.

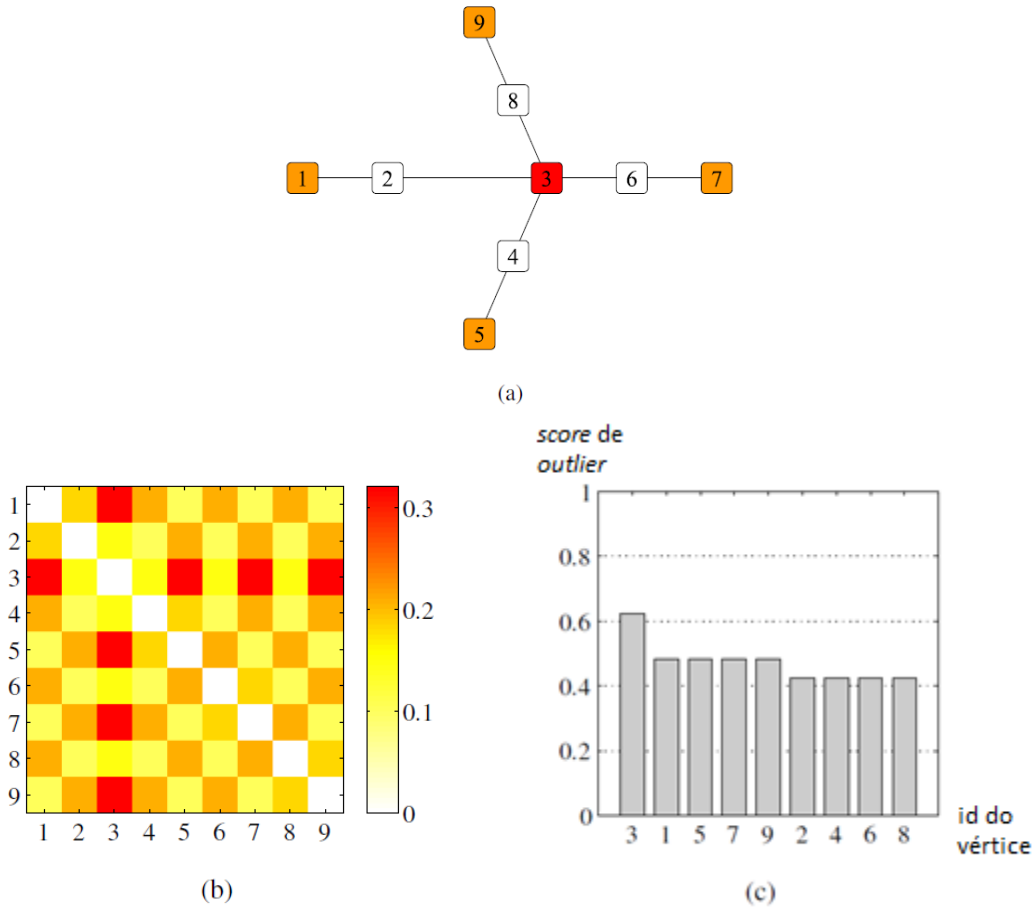


Figura 15: (a) Resultado do método de detecção de *outlier* aplicado em uma cadeia com 9 vértices. Os vértices vermelhos têm o maior *score* de *outlier* seguido pelos laranjas. (b) a matriz de dissimilaridade Λ representada por cores. Vermelho representa o valor mais alto e branco o mais baixo. (c) *Rank* de *outlier*.

A partir dos resultados das últimas duas simulações, é interessante observar que o critério de *outlier* não é modificado, mas o método pode detectar diferentes tipos de *outliers* de acordo com a rede de entrada.

Tomando uma rede maior (representada na Figura 16 (a)), o método identifica com maior *score* o vértice central 3, seguido pelos vértices 2, 4, 6 e 8. Os vértices mais distantes aparecem nas últimas posições do ranking, o que permite descartar as hipóteses anteriores. O mapa de cores da Figura 16(b) mostra o índice de dissimilaridade entre cada par de vértices e a Figura 16(c) mostra o *score* de *outlier* de cada vértice. Note que o método identifica vértices diferentes da maioria dos demais vértices da rede e não vértices que sejam diferentes de apenas um nó. Fato é que o vértice 3 tem uma perspectiva diferente de todos os demais na rede, os vértices 2, 4, 6 e 8 compartilham uma mesma perspectiva, e por fim, os vértices 1, 5, 7, 9, 10, 11, 12 e 13 compartilham outra perspectiva.

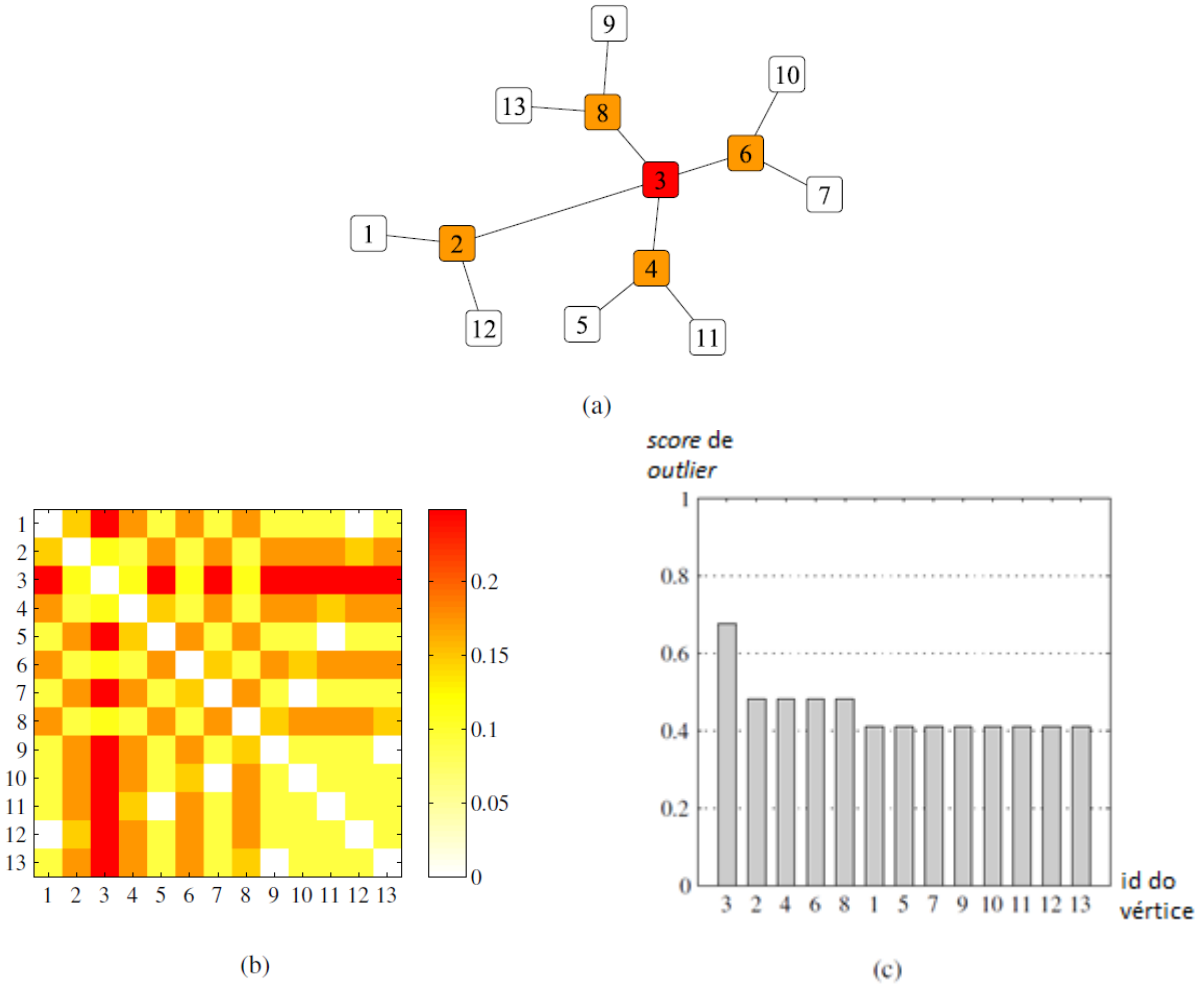


Figura 16: (a) Resultado do método de detecção de *outlier* aplicado em uma cadeia com 13 vértices. Os vértices vermelhos têm o maior *score* de *outlier* seguido pelos laranjas. (b) a matriz de dissimilaridade Λ representada por cores. Vermelho representa o valor mais alto e branco o mais baixo. (c) *Rank* de *outlier*.

Assim, podemos inferir que o método identifica os vértices que possuem uma perspectiva diferenciada como os vértices mais singulares na rede. Não apenas uma perspectiva da distância, mas uma perspectiva singular, diferente da perspectiva de todos os demais vértices.

4.4 Detecção de *outliers* em redes reais

O método de detecção de *outlier* foi aplicado em algumas redes reais. Para cada experimento, selecionamos os 10 vértices com maior *score* de *outlier*. O grau de cada vértice é mostrado também, pode-se observar que não existe relação entre este e o *score* de *outlier*.

A. Rede clube de Karate

A primeira rede analisada foi a rede clube de karate, registrada por Zachary (Zachary, 1977). A rede karate trata-se de uma rede social que modela as iterações entre membros de um clube de karate e possui 34 vértices e 77 arestas. Após a demissão do principal instrutor do clube (vértice 34), a maior parte dos alunos transferiu-se para a nova academia do instrutor, enquanto a minoria optou permanecer na primeira academia. O administrador da primeira academia está representado no vértice 1 da rede. O autor registra também que o que resultou na transferência da maioria dos alunos não foi apenas sua ligação com o instrutor, mas também sua ligação com um amigo do instrutor (vértice 33) muito influente na academia.

A rede clube de karate pode ser visualizada na Figura 17, assim como os 10 vértices com maior *score* de *outlier*. A Tabela 2 apresenta o *score* de *outlier* dos 10 primeiros vértices e seus graus. Observa-se que os vértices 12 e 17 possuem pouca influência sobre os demais, enquanto os vértices 34, 1 e 33 representam os vértices centrais na rede. Eles correspondem ao instrutor da academia, ao amigo do instrutor e ao administrador, respectivamente. O vértice 3 também se destaca pois apresenta ligações com membros de ambas as comunidades, sendo considerado diferente dos demais vértices e logo um *outlier*.

B. Sub-rede de colaboração científica

A segunda rede analisada foi a rede de colaboração entre cientistas que estudam redes complexas. Esta rede compilada por Newman (2006) modela a colaboração entre cientistas com base em suas publicações. Dessa rede optamos por utilizar apenas uma sub-rede densamente conectada, que corresponde a rede de colaboração da qual Newman é participante. Essa sub-rede possui 379 vértices e 914 arestas.

A sub-rede com os 10 vértices de maior *score* de *outlier* pode ser visualizada na Figura 18 e conforme mostra a Tabela 3, nenhum destes vértices é muito colaborativo. Autores mais colaborativos como H. Jeong (vértice 5), A. L. Barabási (vértice 4) e M. Newman (vértice 26) aparecem respectivamente na posição 50, 61 e 62 no ranking de *outlier*. Nessa rede há muitos indivíduos com apenas uma conexão, porém a maioria deles está conectada com indivíduos com muitas ligações. Isso leva-nos a inferir que os vértices mais singulares são aqueles que têm colaboração limitada e cujos colaboradores também têm um baixo nível de colaboração.

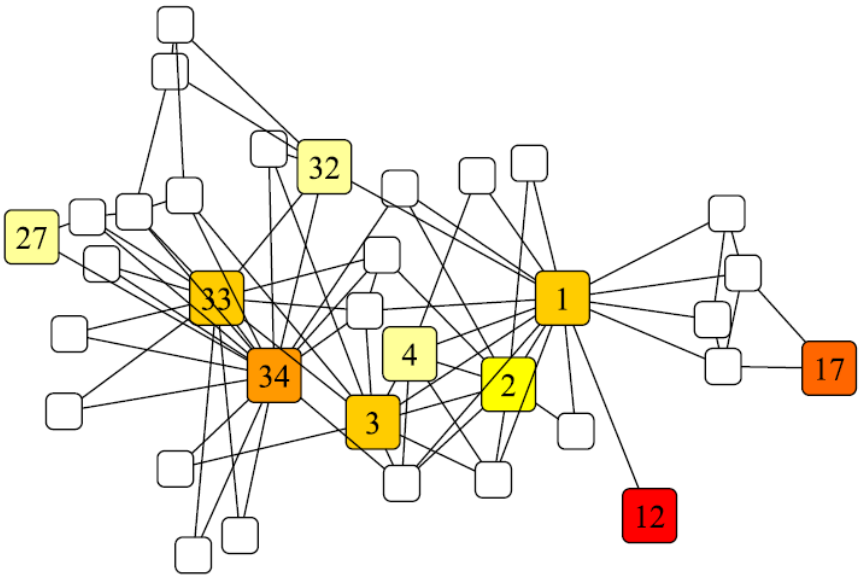


Figura 17: Método de detecção de *outlier* aplicado à rede clube de karate (Zachary, 1977). Os 10 vértices com maior *score* de *outlier* aparecem colorido, quanto mais vermelho, maior seu *score* de *outlier*.

Tabela 2. *Rank* dos 10 vértices com maior *score* de *outlier* na rede clube de karate

	Vértice	σ	grau
1º	12	0,1625	1
2º	17	0,1099	2
3º	34	0,0738	17
4º	1	0,0729	16
5º	3	0,0717	10
6º	33	0,0692	12
7º	2	0,0656	9
8º	27	0,0598	2
9º	32	0,0570	6
10º	9	0,0566	5

Tabela 3. *Rank* dos 10 vértices com maior *score* de *outlier* na sub-rede de colaboração científica

	Vértice	σ	grau
1º	37	0,1057	1
2º	209	0,0674	1
3º	305	0,0650	2
4º	38	0,0649	3
5º	48	0,0539	1
6º	281	0,0520	1
7º	138	0,0500	1
8º	272	0,0489	2
9º	377	0,0485	3
10º	376	0,0485	3

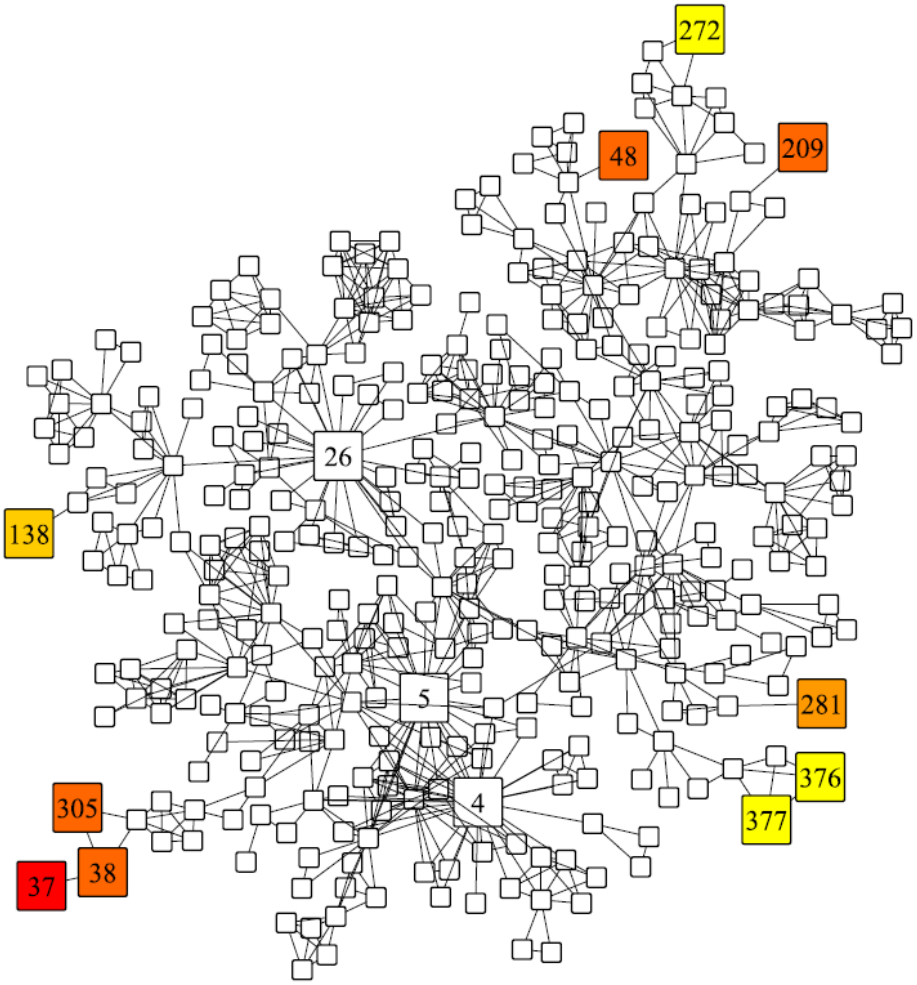


Figura 18: Resultado do método de detecção de *outlier* aplicado à sub-rede de colaboração científica. Os 10 vértices com maior *score* de *outlier* da rede aparecem colorido. Quanto mais vermelho o vértice, maior seu *score* de *outlier*.

4.5 Discussão do método

O método de detecção de *outliers* em redes complexas apresentado é baseado na medida de distância da caminhada aleatória de uma partícula Browniana e no índice de dissimilaridade apresentado por Zhou (2003b). A medida de distância calcula a perspectiva que um vértice tem dos demais na rede, enquanto que a dissimilaridade mede a diferença de perspectiva da rede entre cada par de vértices. Diferente de muitas medidas de *outliers*, essa medida combina informação local e global da rede por meio da natureza da caminhada aleatória, sendo então possível identificar diferentes tipos de vértices *outliers*, conforme mostrado nas simulações das Seções 4.3 e 4.4.

Os resultados experimentais sugerem que os vértices *outliers* em uma rede, não são apenas aqueles distantes do centro, mas aqueles que possuem perspectivas diferentes de toda a rede. Acredita-se que o método proposto é um bom estimador de vértice com uma função especial na rede, fornecendo uma nova perspectiva para os estudos de *outliers*.

Capítulo 5

Caracterização de classes via otimização em redes complexas

O desempenho de um classificador depende de diferentes fatores. Um fator muito importante se refere às características dos dados a serem classificados. Na prática, nenhum classificador é o melhor em todos os problemas dados, fenômeno que pode ser explicado pelo teorema *no free lunch* (Worlpert e Macready, 1997).

Como nem sempre se tem conhecimento prévio sobre a distribuição de um conjunto de dados ou sobre as características das classes, procuramos neste trabalho propor medidas que auxiliem na caracterização de classes de dados. Executamos o algoritmo proposto em alguns cenários de dados artificiais, buscando analisar a mistura dos dados, alterando-se a forma dos dados e a quantidade de classes.

No que segue, a Seção 5.1 descreve o método proposto, apresentando a função de energia utilizada, as medidas de extensão e pureza e o algoritmo com detalhes da implementação. A Seção 5.2 mostra os resultados obtidos a partir da aplicação do método em redes artificiais, bem como análise dos mesmos. A Seção 5.3 mostra resultados obtidos com a aplicação do método em redes reais. Por fim a Seção 5.4 aborda uma discussão do método proposto.

5.1 Método proposto

A motivação para este trabalho partiu das redes *K*-associados (Lopes et al., 2009). Essas redes são construídas com base em uma medida de pureza e como esta medida não considera o tamanho dos componentes de maneira explícita, ela favorece a formação de muitos componentes pequenos quando se tem um nível alto de mistura nas classes, fazendo com que a classificação possa ter problemas devido a ruídos nos dados de treinamento, caso os dados

que ficaram sozinhos são considerados na classificação. E no caso destes dados serem desconsiderados, pode-se ter uma grande quantidade de dados inutilizados. Uma boa classificação depende de um equilíbrio entre a pureza e o tamanho dos componentes, por essa razão, buscou-se desenvolver uma técnica considerando não apenas o fator de *pureza*, mas também *extensões* de classes formadas.

No método proposto, uma rede é construída a partir de um conjunto de dados proposicionais representados no formato atributo-valor. A rede é construída com base nas relações de similaridade entre os vértices, de modo que cada vértice irá alterar suas conexões com um *k*-vizinho mais próximo que seja de sua mesma classe, somente se essa alteração maximize a *função de energia* aqui sugerida. A utilização dessa função de energia foi inspirada pelo trabalho de Cancho e Solé (2003), no qual se buscava a minimização da quantidade de arestas e da média do menor caminho.

Procurou-se definir uma função de energia que considere a relação entre a pureza e a extensão da rede, visto que os dois fatores são opostos, pois o aumento da pureza tende a diminuir a extensão e vice-versa. A função de energia é representada pela Equação (55):

$$E = \lambda d + (1 - \lambda)p \tag{55}$$

na qual, d representa a extensão e p representa a pureza, tal que p e $d \in [0,1]$. A partir das medidas de pureza e extensão, investigamos o comportamento delas na rede alterando gradativamente o peso de cada uma na função de energia, para tal um parâmetro $\lambda \in [0,1]$ foi introduzido.

Para a proposta considerada, a medida de extensão é computada do seguinte modo: o caminho mínimo é calculado para *N* vértices de cada componente formado na rede. O valor máximo encontrado para cada componente *C* representa o diâmetro do mesmo. Podemos obter assim a média de diâmetro dos componentes:

$$\langle d \rangle = \left(\sum_{i=1}^{C} d_i\right)/C \tag{56}$$

Para que $\langle d \rangle$ permaneça no intervalo entre 0 e 1, é normalizado novamente pelo maior diâmetro possível de ser encontrado quando consideramos a maior classe como um grafo linear. Obtemos assim *d*, que representa a extensão da rede.

$$d = \langle d \rangle / d_{max} \tag{57}$$

A pureza de um vértice *i* se refere à relação entre o total de ligações que um vértice estabeleceu e o número máximo que ele poderia ter estabelecido. Seja g_i o grau de saída do vértice v_i, *k* um parâmetro para controlar o número de vizinhos usados na construção da rede.

A Equação (58) corresponde a relação entre o número de conexões entre o vértice v_i e os vértices em seu próprio componente.

$$p_i = g_i/k \tag{58}$$

A medida *p* se refere a pureza de toda a rede, sendo a média da pureza dos vértices.

$$p = (\sum_{i=1}^{N} p_i)/N \tag{59}$$

Algoritmo 1

Entrada: Conjunto de vértices: $V = v_1, \dots, v_n$
Conjunto de classes: $L = classe(v_1), \dots, classe(v_n)$
Parâmetro $\lambda \in [0,1]$
Saída: Rede gerada: R

1) **Para** cada vértice v_i de V
Para cada vértice v_j de V
S = Calcula_similaridade (v_i, v_j);
S = Ordena_crescente (S);
R = V;

2) p = Calcula_pureza (R);
d = Calcula_extensão (R);
E = Calcula_Energia (p, d, λ);

3) **Para** *cont* = 1 até N
R_Δ = Modifica_rede (R, S, V, L);
p_Δ = Calcula_pureza (R_Δ);
d_Δ = Calcula_extensão (R_Δ);
E_Δ = Calcula_Energia $(p_\Delta, d_\Delta, \lambda)$;
Se $E_\Delta > E$
$E = E_\Delta$;
$R = R_\Delta$;

3.1) Modifica_rede (R, S, V, L)
v_i = *random*();
k = *random*();
Se $(k < n)$
v_k = Busca_vertice (S);
Se $(classe(v_i) == classe(v_k))$
Se (!Existe_aresta(v_i, v_k))
$R_\Delta \leftarrow R_\Delta \cup$ Insere_aresta (V);
Senão
$R_\Delta \leftarrow R_\Delta -$ Remove_aresta (V);
4) **Retorna** R.

O Algoritmo 1 descreve com detalhes a implementação do método. A entrada do algoritmo é um conjunto de dados representados na forma atributo-valor, com a respectiva classe associada e um parâmetro $\lambda \in [0,1]$, que controlará o peso dado à pureza e à extensão.

No passo 1, o algoritmo calcula a partir dos dados de entrada a matriz de similaridade, nesse caso foi calculada a distância euclidiana entre todos os pares de vértices. Posteriormente cada linha dessa matriz é ordenada em ordem crescente. A rede *R* é inicializada, de modo que cada objeto de dado é representado em um vértice.

No passo 2, são calculadas as medidas de extensão da rede, Equação (57), de pureza, Equação (59), a partir destas é realizado o cálculo da energia, Equação (55), com o valor de λ fornecido inicialmente.

O passo 3 é repetido *N* vezes. Outra opção que poderia ser utilizada como critério de parada, seria executar o algoritmo até que *N* vezes o valor de *E* não sofra alteração.

O passo 3.1 *Modifica-rede* é executado retornando uma nova rede com modificações nas ligações para um vértice *i*. A partir da nova rede, as medidas de pureza e extensão são recalculadas e a função de energia é reavaliada para estas novas medidas. Caso o valor da nova função de energia seja maior que o anterior, a rede antiga é substituída pela nova rede que sofreu modificações.

No passo 3.1 um vértice *i* é escolhido aleatoriamente para sofrer alteração nas ligações. Também é escolhido aleatoriamente um vértice *k*, que esteja dentro do conjunto dos *n*-vizinhos mais próximos do vértice *i*, para se conectar/desconectar com este. Nos testes realizados, o vértice *k* é sorteado em um intervalo entre 1 e 5.

Destaca-se que o vértice *k* é selecionado aleatoriamente, porém o 1-vizinho tem uma probabilidade maior de ser escolhido, que o 2-3...*n* -vizinho respectivamente. Do mesmo modo, o 2-vizinho tem uma probabilidade menor que o 1 de ser selecionado, porém sua probabilidade é maior que o 3-4... *n*-vizinho respectivamente. E assim por diante. O objetivo disto é fazer com que os vértices mais similares de um vértice *i* tenham preferência nas conexões.

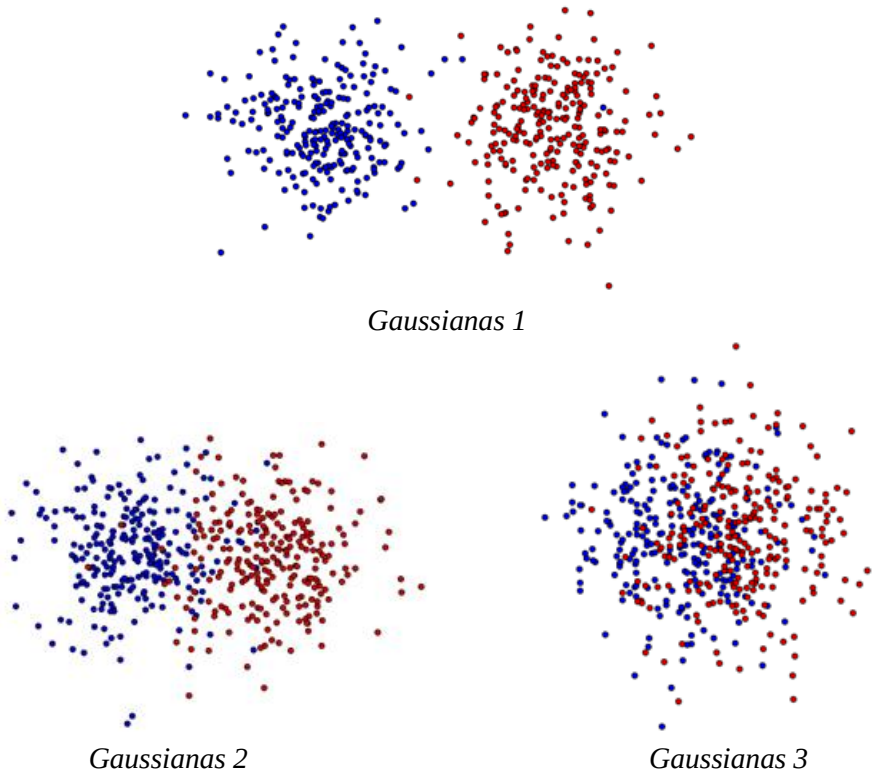
Caso o vértice *k* escolhido não esteja ligado com o vértice *i*, uma ligação entre eles será estabelecida se ambos pertencerem a mesma classe. Além disso, o vértice *i* irá estabelecer ligações com todos os *k*-vizinhos menores que *k*. Porém, se o vértice *k* já estiver conectado com o vértice *i*, então esta conexão é removida, bem como todas as conexões existentes entre o vértice *i* e os vértices maiores que *k*.

Essa nova rede irá conter alterações nas ligações para um dado vértice *i* e será retornada para o Passo 3. Por fim, o Passo 4 retorna a rede final, a qual apresenta a maximização da função de energia, para um dado valor de λ.

5.2 Simulações em redes artificiais

O algoritmo foi executado para alguns conjuntos de dados artificiais, a fim de analisar-se seu comportamento em bases cujas características já eram previamente conhecidas.

Inicialmente foi testado para um conjunto de dados com distribuição gaussiana, de modo que, para cada conjunto de dados aumentava-se gradativamente a mistura dos elementos (Figura 19):



Gaussianas 1

Gaussianas 2

Gaussianas 3

Figura 19: Base de dados *Gaussianas 1-2-3*. Cada figura representa dois conjuntos de dados com 250 elementos e distribuição gaussiana, a cor vermelha representa uma classe e a cor azul outra.

Na Figura 20 são mostradas as redes finais formadas para as bases de dados *Gaussianas 1-2-3*, após a execução do Algoritmo 1 para os valores de λ igual a 0 e 1. Nota-se que quando $\lambda = 0$ apenas a pureza está sendo levada em conta e quando $\lambda = 1$ apenas a extensão é considerada.

As redes *A* e *B* foram formadas a partir da base de dados *Gaussianas 1*. Nota-se que ambas as redes formaram poucos componentes, porém na rede *A* há mais conexões entre os vértices e na rede *B* o diâmetro dos componentes é maior que na rede *A*. As redes *C* e *D* foram formadas a partir da base de dados *Gaussianas 2*. Como o nível de mistura foi aumentado um número maior de componentes é formado, porém a rede *C* apresenta mais componentes que a rede *D*. As redes *E* e *F* foram formadas a partir da base de dados *Gaussianas 3* cujo nível de mistura está maior que das bases *Gaussianas 1* e *2*. Com isso, o número de componentes formados é bem alto quando $\lambda = 0$ (rede *E*), pois como cada vértice apresenta muitos

vizinhos com classes diferentes da sua, acaba estabelecendo ligações com pouco deles para permanecer como uma pureza alta. Quando $\lambda = 1$, o número de componentes formados na rede *F* diminui com relação a rede *E*.

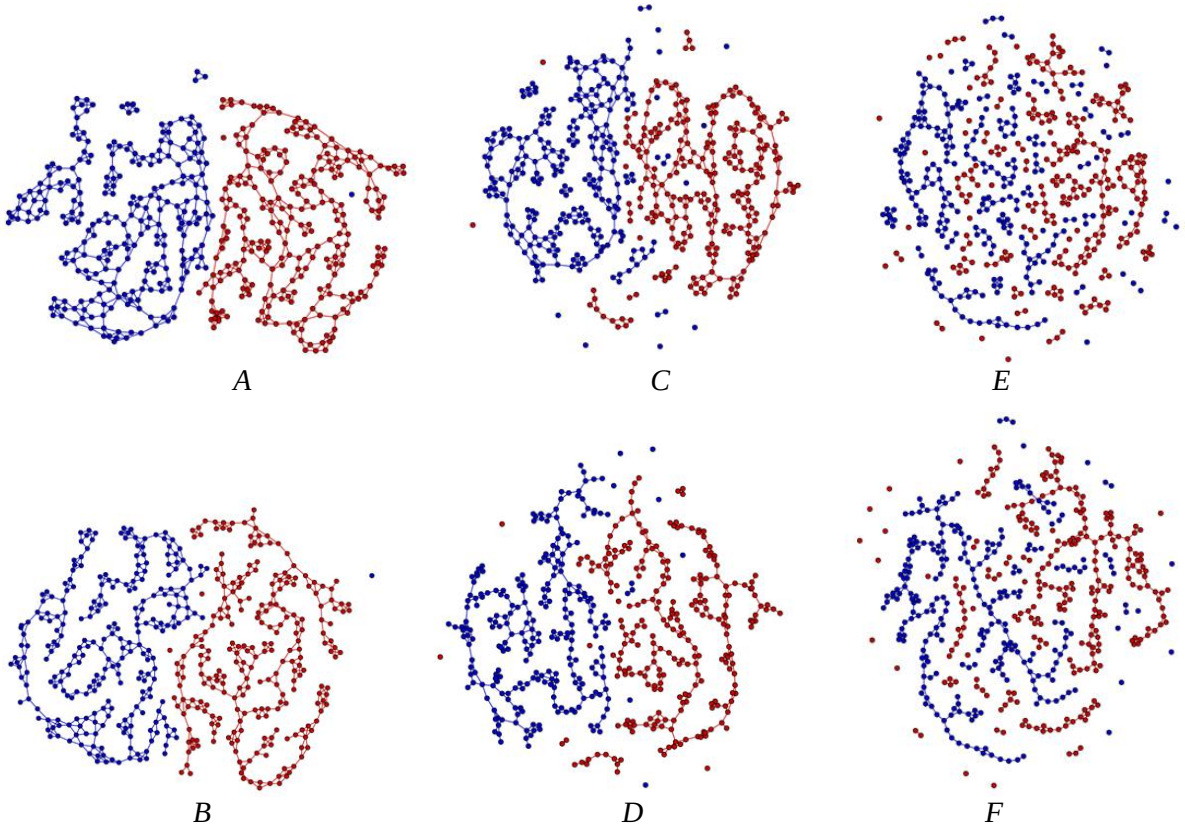


Figura 20: *A* e *B*: redes finais formadas para a base de dados *Gaussianas 1*. *C* e *D*: Redes finais formadas para a base de dados *Gaussianas 2*. *E* e *F*: Redes finais formadas para a base de dados *Gaussianas 3*. A rede *A*, *C* e *E* foram construídas com λ igual a 0 e a rede *B*, *D* e *F* foram construídas com λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo1.

A Figura 21 mostra a pureza das redes finais geradas quando a função de energia é maximizada em alguns valores de λ para as bases de dados *Gaussianas 1-2-3*. Nota-se que o valor da pureza para a base *Gaussianas 3* é menor que da base *Gaussianas 2* e *Gaussianas 1* respectivamente. Isso porque a base *Gaussianas 3* apresenta alto nível de mistura entre os elementos de diferentes classes, e com isso componentes "menos puros" são formados. Ou seja, cada vértice estabelece ligações com um número menor de vizinhos que estava sendo considerado no momento.

Além disso, a pureza em cada base se mantém praticamente constante, diminuindo levemente conforme λ se aproxima de 1. Isso acontece, pois os vértices conseguem completar suas ligações tanto para um *k*-vizinho menor como para um *k*-vizinho maior, permanecendo com valor alto de pureza.

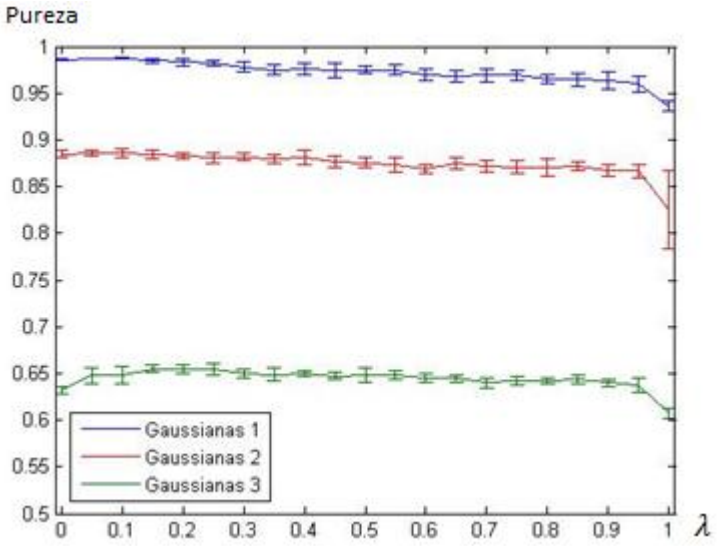


Figura 21: Representação da pureza para as redes finais geradas a partir das bases de dados *Gaussianas 1-2-3*, com média sobre 30 execuções do algoritmo.

A Figura 22 mostra a extensão das redes finais geradas quando a função de energia é maximizada em alguns valores de λ, para as bases de dados *Gaussianas 1-2-3*. Observa-se que o valor da extensão para a base *Gaussianas 3* é menor que da base *Gaussianas 2* e *Gaussianas 1* respectivamente, devido ao nível de mistura apresentado na base *Gaussianas 3*. Com isso, os vértices não conseguem formar um único componente para cada classe, diminuindo o valor para a extensão. Além disso, o valor de extensão aumenta conforme λ se aproxima de 1, já que esta medida passa a ter mais destaque no cômputo da função de energia.

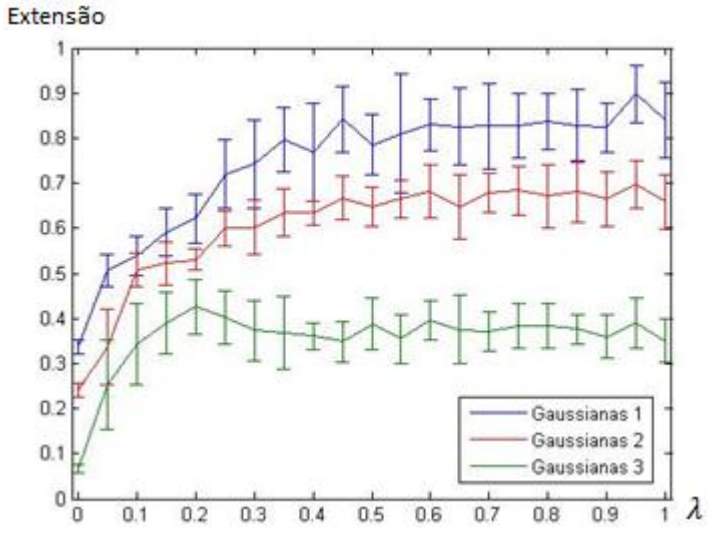


Figura 22: Representação da extensão para as redes finais geradas a partir das bases de dados *Gaussianas 1-2-3*, com média sobre 30 execuções do algoritmo.

Na Figura 23 a função de energia é mostrada nas bases *Gaussianas 1-2-3* para alguns valores de λ. Essa decai conforme λ se aproxima de 1 porque passa a dar mais peso para a extensão e essa medida obtém um valor menor que o da pureza para todas as redes.

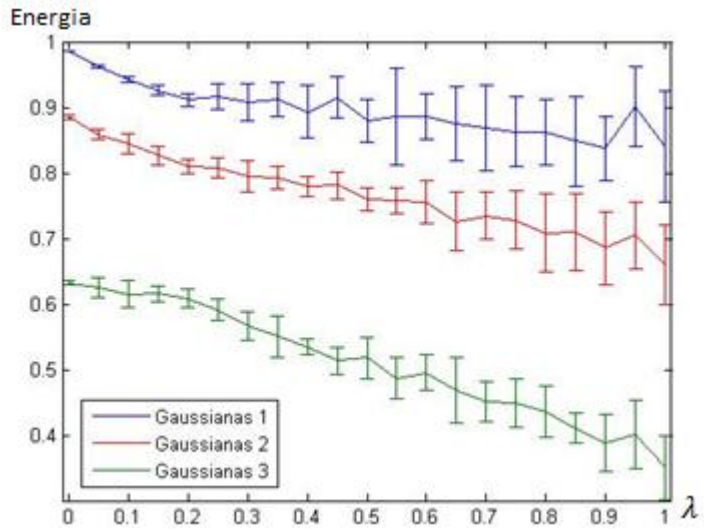


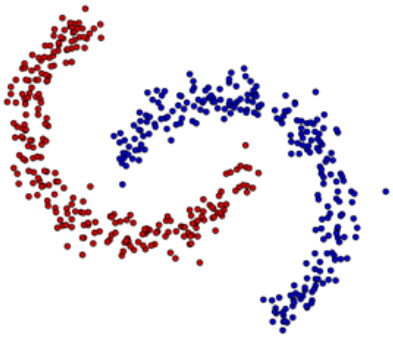
Figura 23: Representação da energia para as bases de dados *Gaussianas 1-2-3*, com média sobre 30 execuções do algoritmo.

Analisando as redes finais formadas na Figura 20, nota-se que conforme a mistura entre dados de diferentes classes aumenta o número de componentes formados na rede também aumenta, tanto para $\lambda = 0$, como para $\lambda = 1$. Porém para $\lambda = 1$ a tendência é de formar menos componentes que para $\lambda = 0$.

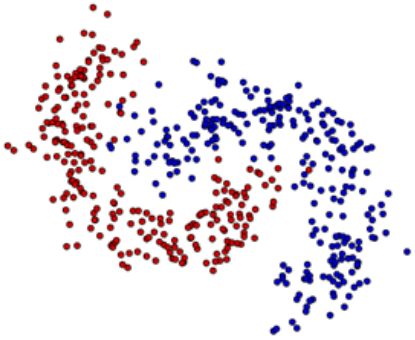
Analisando os resultados para pureza, extensão e energia, nota-se que as três medidas obtêm valores mais baixos conforme o nível de mistura aumenta, de modo que estas medidas podem ser utilizadas para caracterizar a mistura nas classes dos dados. Além disso, observa-se que conforme a extensão das redes aumenta a pureza diminui levemente, indicando que seria possível utilizar uma rede formada para um valor maior de λ, já que esta teria um número menor de componentes formados, e isto poderia ser mais interessante para o processo de classificação.

Posteriormente, o algoritmo foi executado para um conjunto de dados *banana shaped*, aumentando-se gradativamente a mistura dos elementos para cada conjunto de dados (Figura 24).

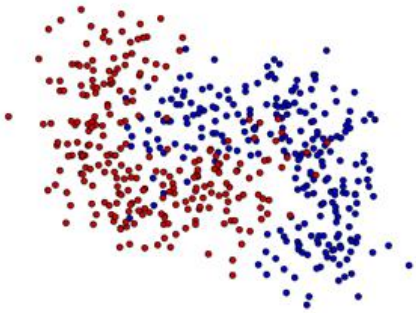
Na Figura 25 são mostradas as redes finais formadas para as bases de dados *Bananas 1-2-3*, após a execução do Algoritmo 1 para os valores de λ igual a 0 e 1. Observa-se que os resultados obtidos para os dados das bases *Bananas 1-2-3* se assemelham aos resultados das bases *Gaussianas 1-2-3*, de modo que conforme o nível de mistura aumenta nas bases, um número maior de componentes é formado na rede, principalmente quando $\lambda = 0$. Porém, o número de componentes formados para a base *Bananas 3* acaba sendo menor do que a base *Gaussianas 3*, já que a base *Bananas 3* apresenta um nível de mistura menor.



Bananas 1

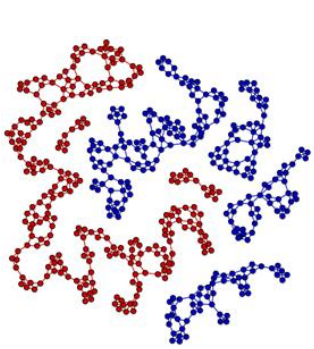


Bananas 2

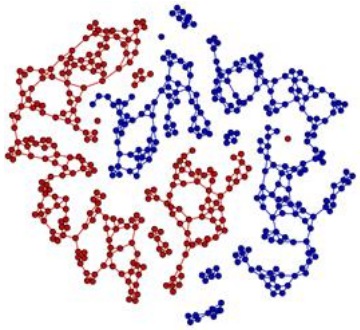


Bananas 3

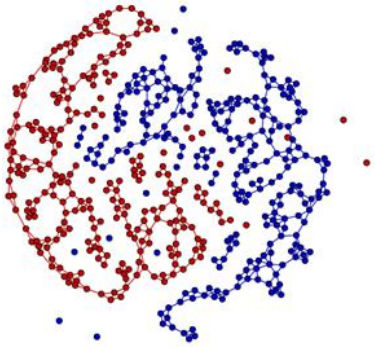
Figura 24: Base de dados *Bananas 1-2-3*. Cada figura representa dois conjuntos de dados com 250 elementos e forma banana, a cor vermelha representa uma classe e a cor azul outra.



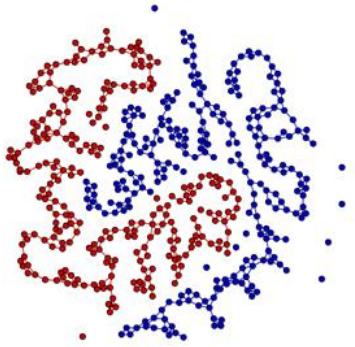
A



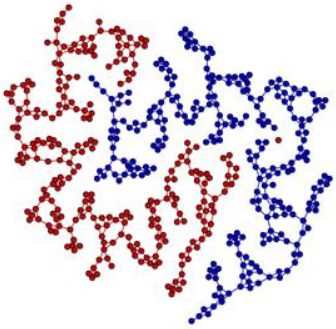
C



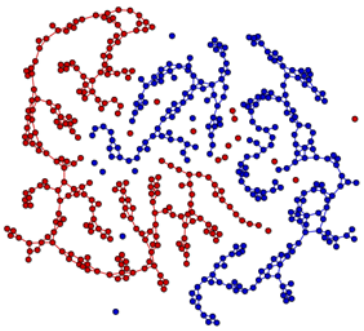
E



B



D



F

Figura 25: *A* e *B*: redes finais formadas para a base de dados *Bananas 1*. *C* e *D*: Redes finais formadas para a base de dados *Bananas 2*. *E* e *F*: Redes finais formadas para a base de dados *Bananas 3*. A rede *A*, *C* e *E* foram construídas com λ igual a 0 e a rede *B*, *D* e *F* foram construídas com λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo1.

A Figura 26 mostra os resultados para pureza nas bases *Bananas 1-2-3* quando a função de energia é maximizada em alguns valores de λ. A análise da pureza obtida nas redes indica bastante semelhança com os resultados obtidos para as bases *Gaussianas 1-2-3*, ou seja, conforme a mistura aumenta entre as classes, o valor da pureza nas redes diminui.

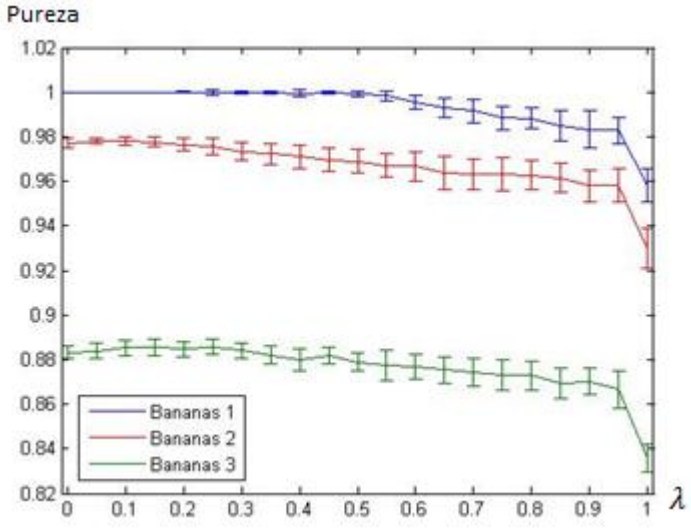


Figura 26: Representação da pureza para as redes finais geradas a partir das bases de dados *Bananas 1-2-3*, com média sobre 30 execuções do algoritmo.

A Figura 27 mostra os resultados para extensão nas bases *Bananas 1-2-3* quando a função de energia é maximizada em alguns valores de λ. Observa-se, contudo, que o valor da extensão para a base *Bananas 1* é menor do que as bases *Bananas 2-3*. Diferindo do comportamento apresentado na base *Gaussianas*.

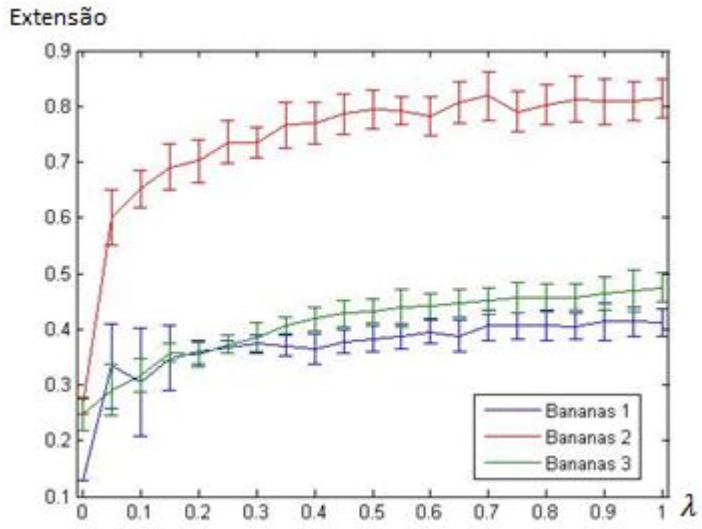


Figura 27: Representação da extensão para as redes finais geradas a partir das bases de dados *Bananas 1-2-3*, com média sobre 30 execuções do algoritmo.

A Figura 28 mostra os resultados para energia nas bases *Bananas 1-2-3* variando-se os valores de λ. A energia da base *Bananas 1* apresenta esse comportamento, devido ao fato de ter a maior pureza e a menor extensão com relação as redes das outras bases.

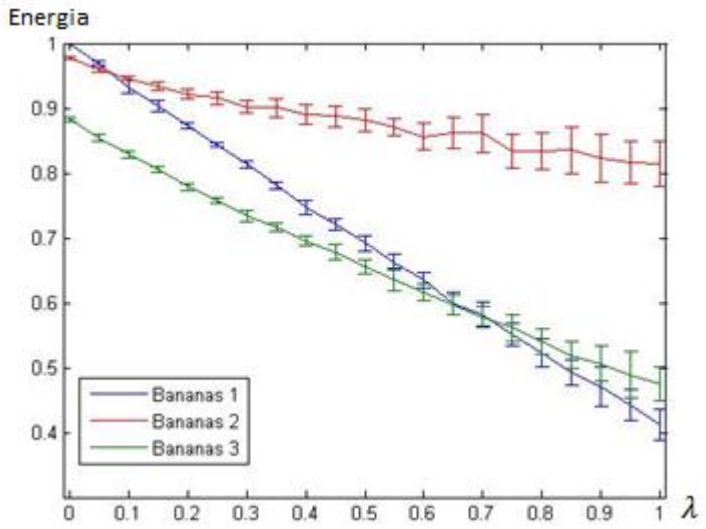


Figura 28: Representação da energia para as redes finais geradas a partir das bases de dados *Bananas 1-2-3*, com média sobre 30 execuções do algoritmo.

O algoritmo também foi executado para um conjunto de dados com distribuição gaussiana, de modo que os dados de uma classe estão mais agrupados (classe azul) e os dados de outra classe estão mais dispersos (classe vermelha), além disso, as duas bases de dados afastam-se gradativamente (Figura 29).

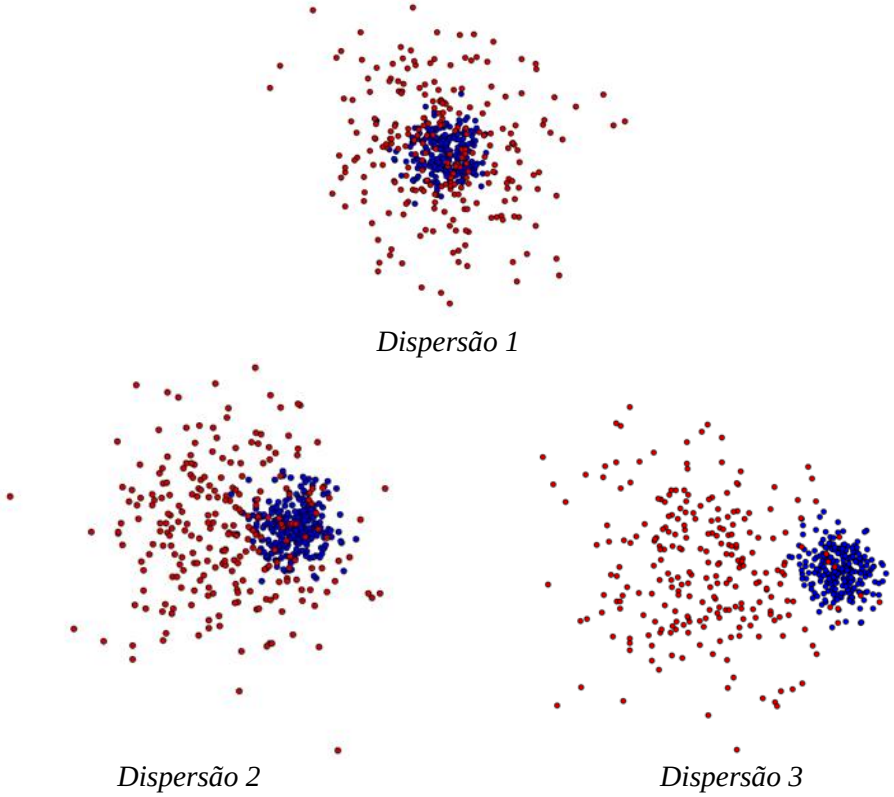


Figura 29: Base de dados *Dispersão 1-2-3*. Cada figura representa dois conjuntos de dados com distribuição gaussiana e 250 elementos cada, tal que a cor vermelha representa uma classe e a cor azul outra.

Na Figura 30 são mostradas as redes finais formadas para as bases *Dispersão 1-2-3*, para os valores de λ igual a 0 e 1. As redes *A* e *B* foram formadas a partir da base de dados *Dispersão 1*. Ambas as redes apresentam poucos componentes formados, apesar da mistura entre as classes. As redes *C* e *D* foram formadas a partir da base de dados *Dispersão 2*. Para esta base mais componentes foram formados, principalmente quando $\lambda = 0$, mas quando $\lambda = 1$ esta quantidade diminui. As redes *E* e *F* foram formadas a partir da base de dados *Dispersão 3*, cujas classes estão mais afastadas, com isso as redes formadas tanto para a classe vermelha quanto para a classe azul se tornam bastante semelhantes, tanto para $\lambda = 0$ como para $\lambda = 1$. Porém para $\lambda = 1$, os componentes possuem um diâmetro maior e menos ligações entre os vértices.

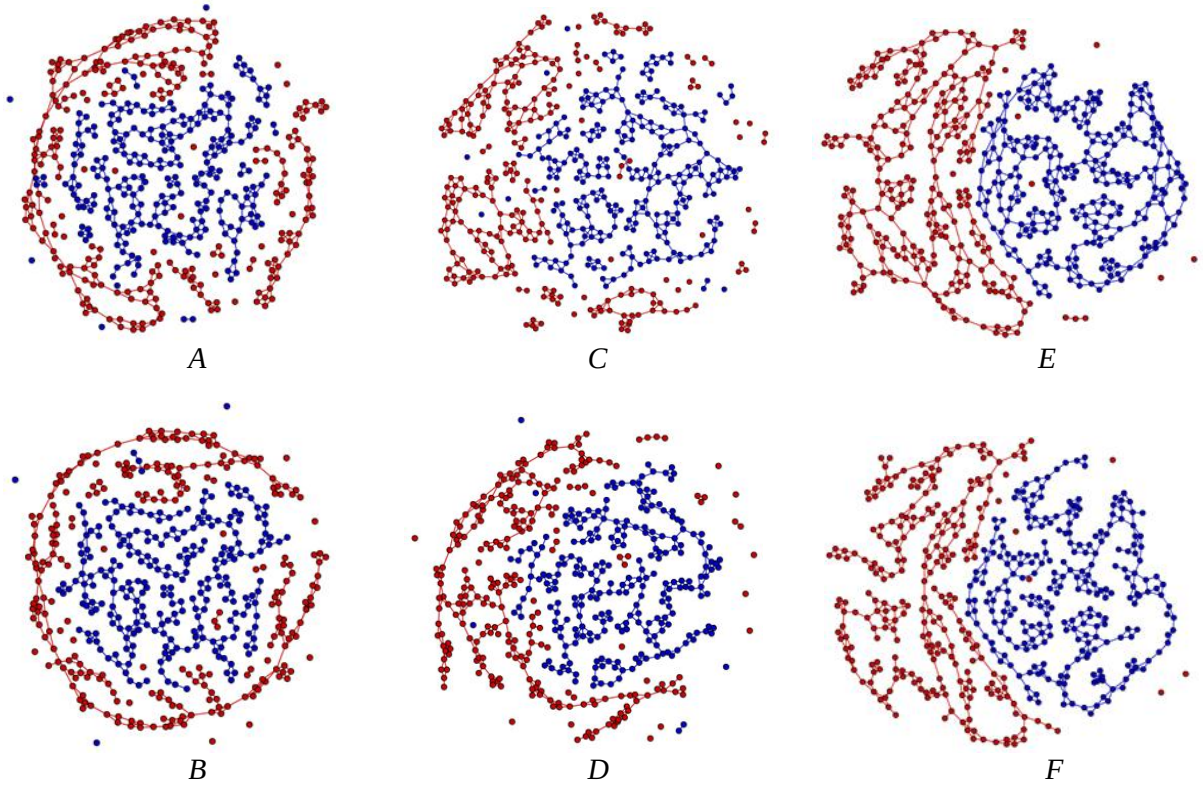


Figura 30: *A* e *B*: redes finais formadas para a base de dados *Dispersão 1*. *C* e *D*: Redes finais formadas para a base de dados *Dispersão 2*. *E* e *F*: Redes finais formadas para a base de dados *Dispersão 3*. A rede *A*, *C* e *E* foram construídas com λ igual a 0 e a rede *B*, *D* e *F* foram construídas com λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo1.

A Figura 31 mostra os resultados para pureza nas bases *Dispersão 1-2-3* quando a função de energia é maximizada em alguns valores de λ. Nota-se que a pureza para a base *Dispersão 1* é menor que para a base *Dispersão 2-3* respectivamente, já que esta base apresenta mais mistura entre as classes. Porém a pureza da base *Dispersão 1* e *2* ficam bem próximas, de modo que a mistura de ambas é bastante equivalente.

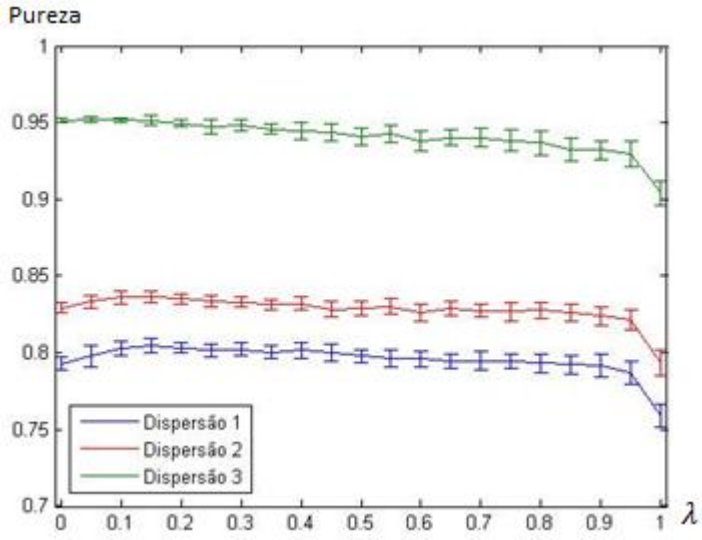


Figura 31: Representação da pureza para as bases de dados *Dispersão 1-2-3*. Média sobre 30 execuções do algoritmo.

A Figura 32 mostra os resultados para extensão nas bases *Dispersão 1-2-3* quando a função de energia é maximizada em alguns valores de λ. Observa-se que a base *Dispersão 1* apresenta valores maiores pois conseguiu formar componentes mais extensos que as demais bases. Seu comportamento se difere ao das bases *Gaussianas* e *Bananas*.

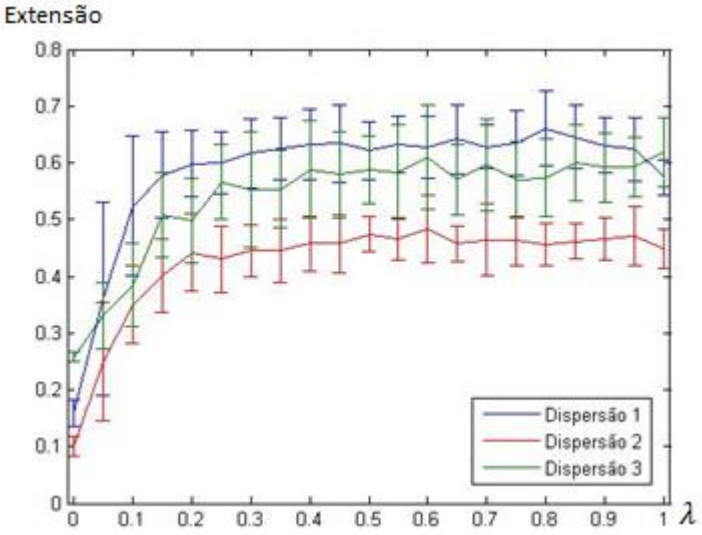


Figura 32: Representação da extensão para as bases de dados *Dispersão 1-2-3*. Média sobre 30 execuções do algoritmo.

A Figura 33 mostra os resultados para a energia nas bases *Dispersão 1-2-3* quando esta é maximizada em alguns valores de λ. A energia da base *Dispersão 1* permeia a energia das outras duas bases, pelo fato da base *Dispersão 1* ter a menor pureza e a maior extensão que as bases *Dispersão 2* e *3*.

Analisando os resultados para a pureza nas bases testadas (*Gaussianas*, *Bananas* e *Dispersão*) nota-se que todos reproduzem um comportamento semelhante. Para extensão o

comportamento das bases *Bananas* e *Dispersão* diferem da base *Gaussianas*, acredita-se que isto aconteça devido ao formato e dispersão diferente dos dados. Com isso, a pureza é mais indicada para caracterizar a mistura entre os dados.

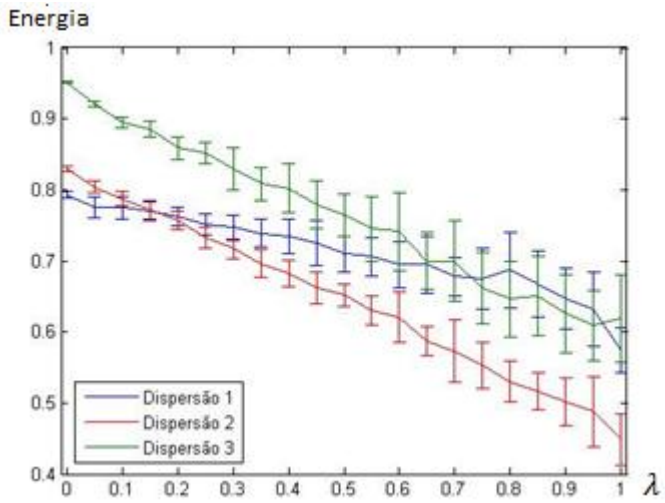
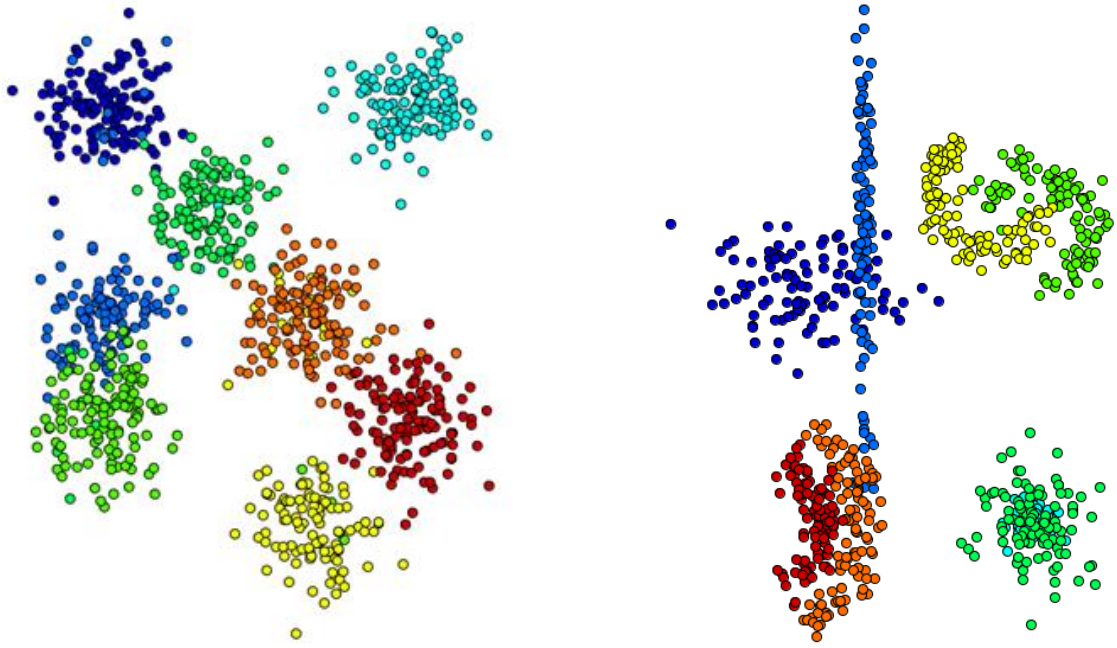


Figura 33: Representação da energia para as bases de dados *Dispersão 1-2-3*. Média sobre 30 execuções do algoritmo.

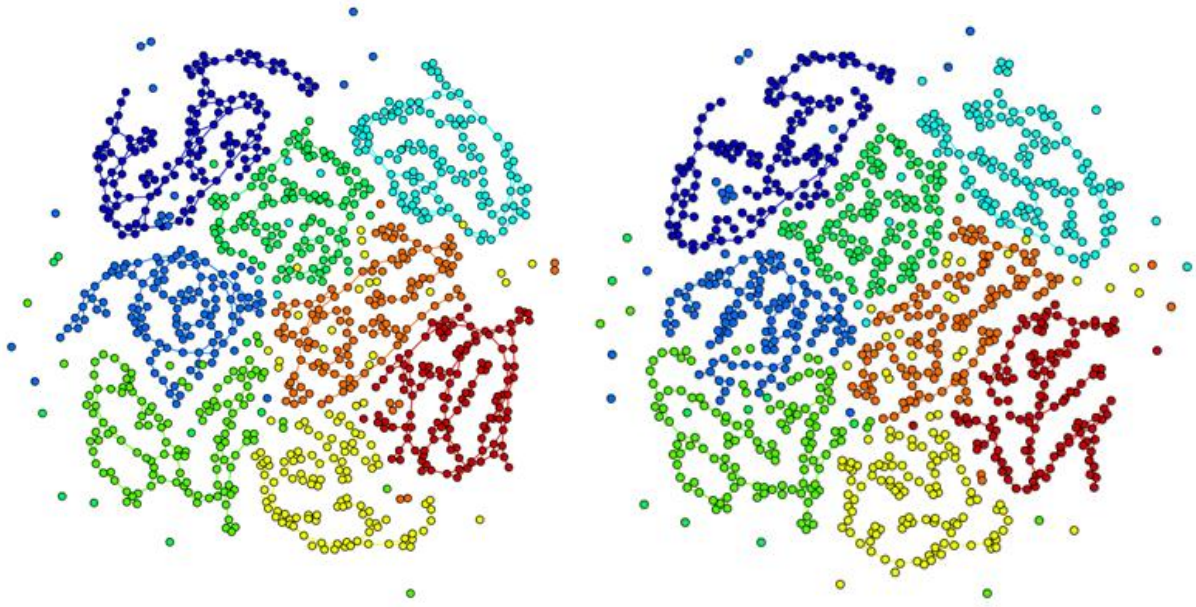
Para o estudo do comportamento da proposta sugerida em bases com uma quantidade maior de classes, o algoritmo foi executado para um conjunto de dados com oito gaussianas e um conjunto de dados *Multiclasse*, o qual é composto por oito classes distintas, sendo que cada classe possui diferentes formatos (Figura 34).



8-Gaussianas *Multiclasse*

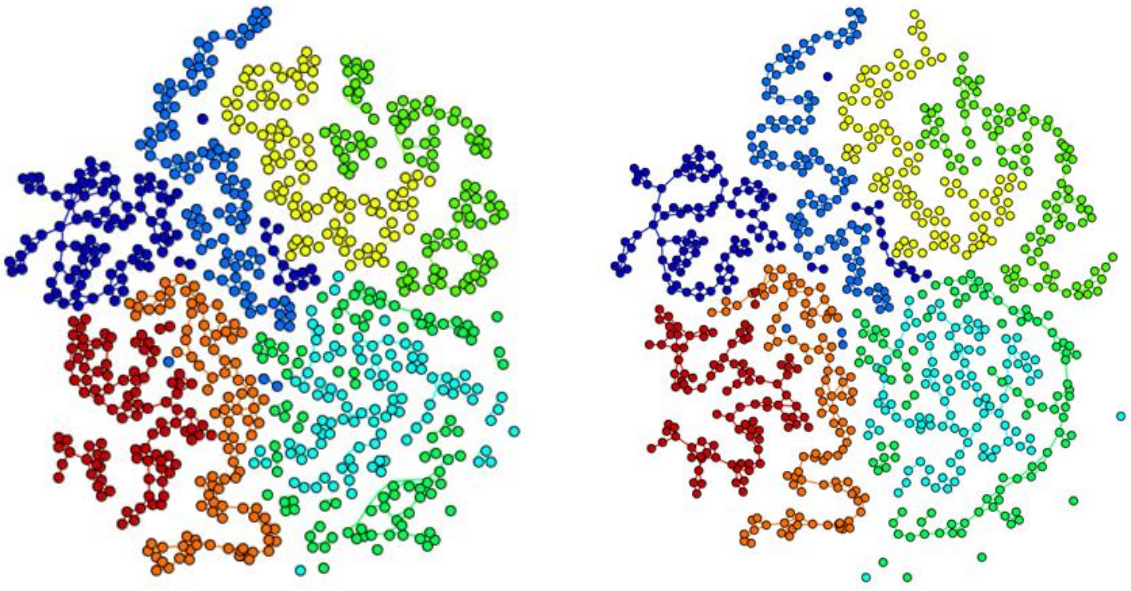
Figura 34: Base de dados *8-Gaussianas* e *Multiclasse*. Cada base possui oito classes de dados com 100 elementos. Cada cor representa uma classe.

Nas Figuras 35 e 36 são mostradas as redes formadas para as bases *8-Gaussianas* e *Multiclasse*, para os valores de λ igual a 0 e 1. As redes se assemelham ao comportamento das demais bases já testadas, de modo que para $\lambda = 1$ menos componentes são formados e o diâmetro dos mesmos aumenta. Observa-se, porém, que para a base *8-Gaussianas*, mesmo quando $\lambda = 1$ há muitos vértices isolados, isto porque há mais mistura nesta base.



A *B*

Figura 35: Redes formadas para a base de dados *8-Gaussianas*, tal que a rede *A* se refere a λ igual a 0 e a rede *B* se refere a λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo1.



A *B*

Figura 36: Redes formadas para a base de dados *Multiclasse*, tal que a rede *A* se refere a λ igual a 0 e a rede *B* se refere a λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo1.

As Figuras 37, 38 e 39 mostram os resultados para pureza, extensão e energia variando-se o valor de λ, para as bases de dados *8-Gaussianas* e *Multiclasse*. Observa-se que os valores alcançados para estas medidas se assemelham para ambas as bases, pois as duas possuem características parecidas, ou seja, são compostas com oito classes de 100 elementos e apresentam classes mais sobrepostas e classes mais afastadas. Os valores para a base *Multiclasse* ficam um pouco acima da base *8-Gaussianas* devido a esta última base apresentar mais mistura entre as classes. Isto indica que as medidas são sensíveis a diferentes níveis de mistura entre as classes.

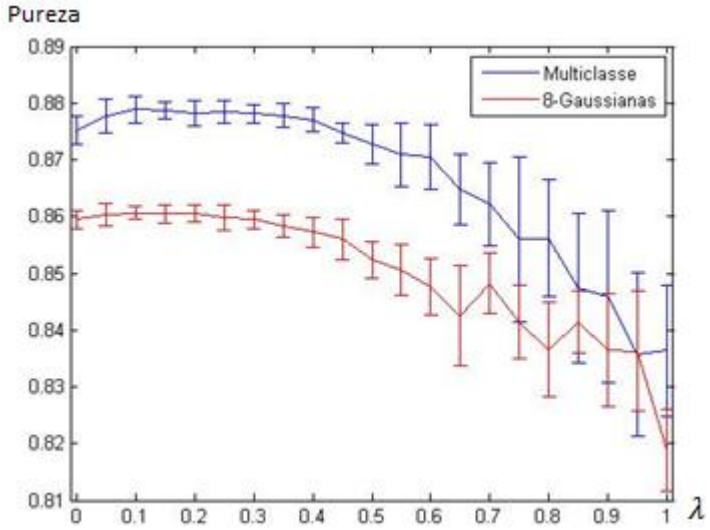


Figura 37: Representação da pureza para as bases de dados *8-Gaussianas* e *Multiclasse*. Média sobre 30 execuções do algoritmo.

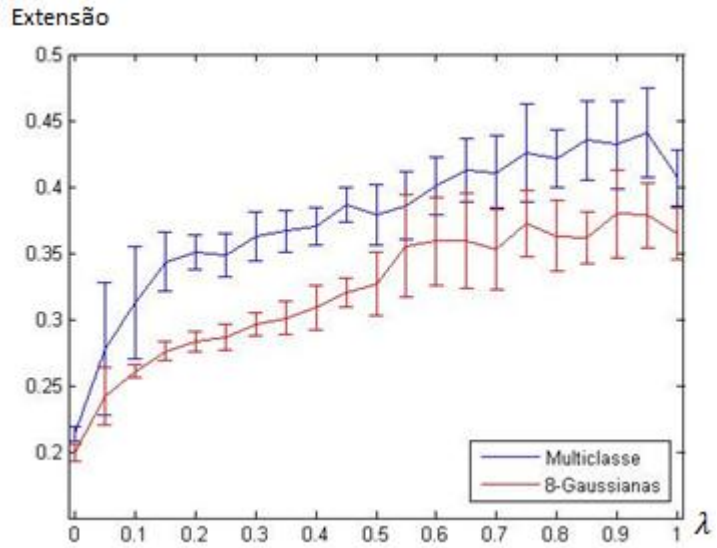


Figura 38: Representação da extensão para as bases de dados *8-Gaussianas* e *Multiclasse*. Média sobre 30 execuções do algoritmo.

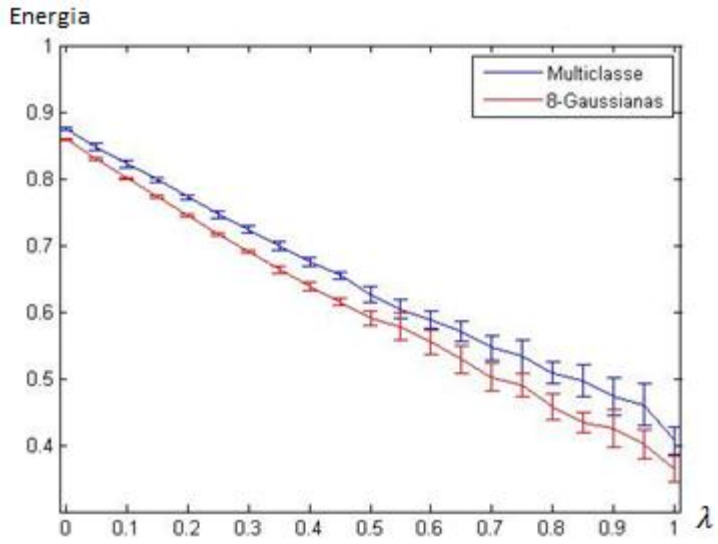


Figura 39: Representação da energia para as bases de dados *8-Gaussianas* e *Multiclasse*. Média sobre 30 execuções do algoritmo.

5.3 Simulações em redes reais

O algoritmo também foi testado para um conjunto de dados reais, executou-se no conjunto de dados Iris, Glass e Zoo, obtidos no repositório UCI[1].

A base de dados Iris (Figura 40) contém 5 atributos e 3 classes com 50 instâncias cada, sendo que cada classe se refere a um tipo de planta Iris. Uma classe é linearmente separada das outras duas, porém a outra não é linearmente separável das demais. A base de dados Glass (Figura 42) contém 10 atributos, 6 classes e 214 instâncias, se refere a um estudo da classificação dos tipos de vidro, o qual foi motivado por investigações criminais, as classes dessa base não são linearmente separável. A base de dados Zoo (Figura 44) contém 18 atributos, sete classes e 101 instâncias, esta base trata de uma repartição dos tipos de animais e as classes também não são linearmente separável.

Para a geração das figuras foi utilizado o *software* PEx[2], com o método de *Sammon´s Mapping* para redução de dimensionalidade.

A Figura 41 mostra as redes formadas após a execução do algoritmo para a base Iris, considerando-se λ igual a 0 e λ igual a 1. Observa-se que para λ igual a 0 (rede *A*) as redes formadas para a classe verde e vermelha não ficaram totalmente conexas, devido a mistura existente nestas duas classes. Já para λ igual a 1 (rede *B*) todas as classes formaram redes conexas, com exceção de um vértice vermelho que não estabeleceu conexões.

[1] http://archive.ics.uci.edu/ml/
[2] http://infoserver.lcad.icmc.usp.br/infovis2/PExDownload

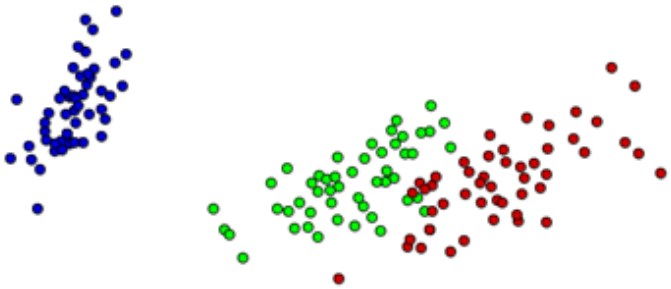
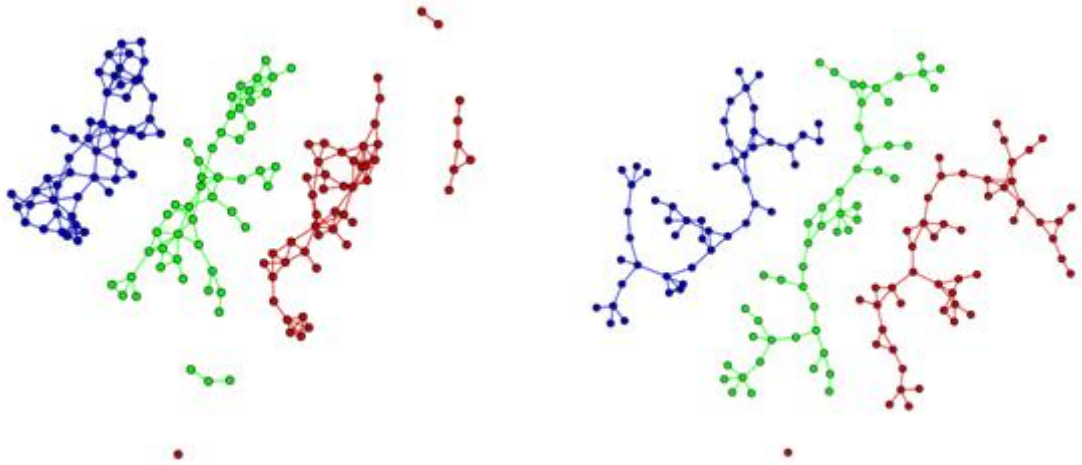


Figura 40: Base de dados Iris.



A *B*

Figura 41: *A*: Rede formada para a base Iris com λ igual a 0. *B*: Rede formada para a base Iris com λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo 1.

A Figura 43 mostra as redes formadas para a base Glass e a Figura 45 mostra as redes formadas para a base Zoo, considerando-se λ igual a 0 e λ igual a 1.

Observa-se que estas bases apresentam mais mistura entre as diferentes classes de dados, devido a isso, as redes formadas tanto para λ igual a 0 (rede *A*), como para λ igual a 1 (rede *B*) ficaram desconexas e com vários componentes.

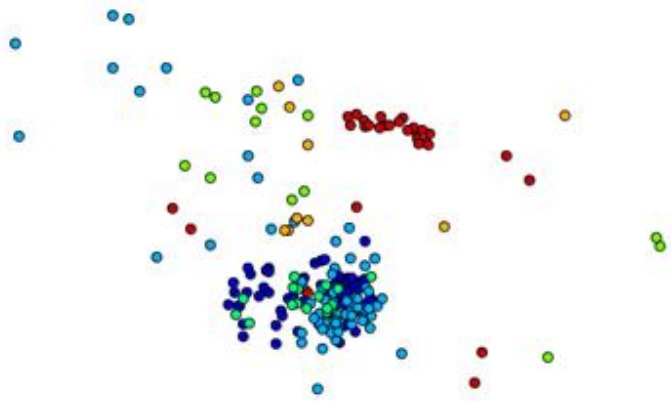
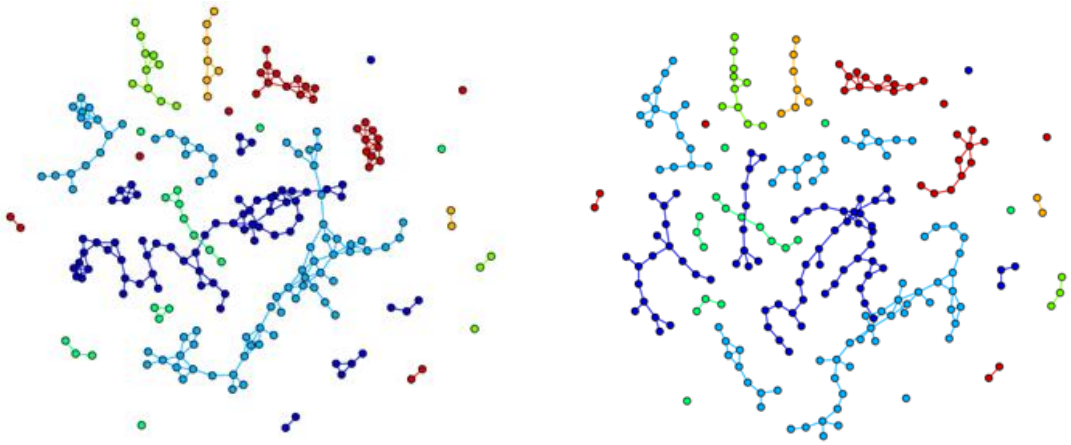


Figura 42: Base de dados Glass.



A *B*

Figura 43: *A*: Rede formada para a base Glass com λ igual a 0. *B*: Rede formada para a base Glass com λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo 1.

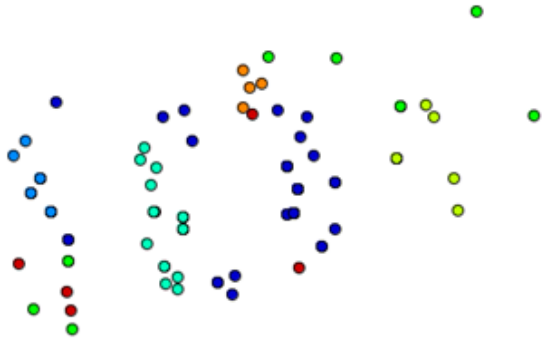
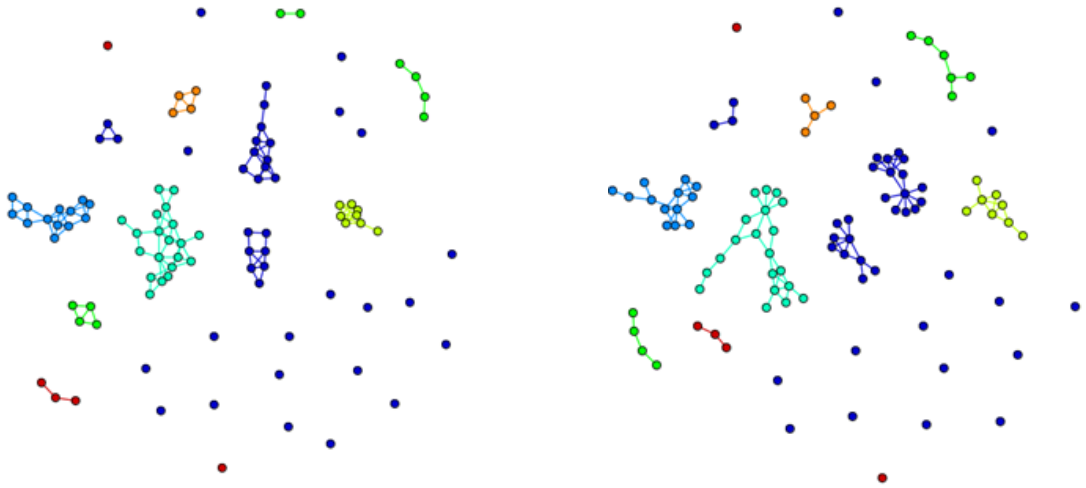


Figura 44: Base de dados Zoo.



A *B*

Figura 45: *A*: Rede formada para a base Zoo com λ igual a 0. *B*: Rede formada para a base Zoo com λ igual a 1. Foram considerados $n = 5$ e $N = 10000$ no Algoritmo 1.

A Figura 46 mostra a pureza das redes finais geradas quando a função de energia é maximizada em alguns valores de λ para as bases de dados Iris, Glass e Zoo. Nota-se que o

valor da pureza para a base Iris assume valores mais próximos de 1, já que esta base apresenta menos mistura entre as classes. As bases Glass e Zoo assumem valores mais baixos já que apresentam mais mistura entre as diferentes classes.

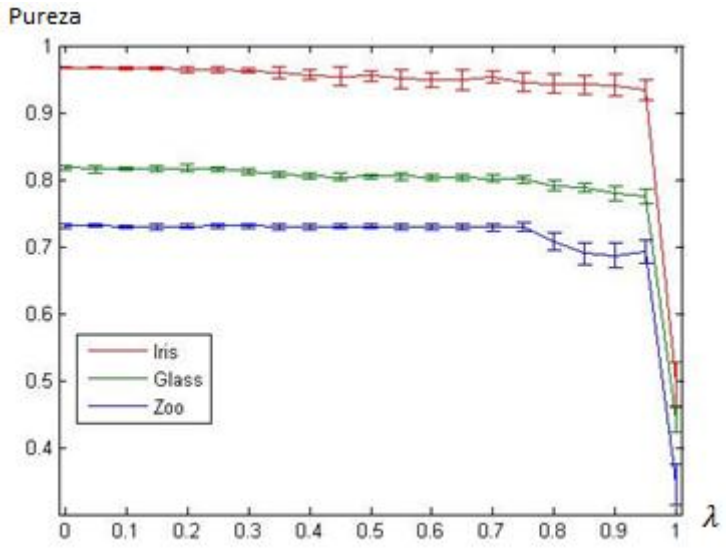


Figura 46: Representação da pureza para as bases de dados Iris, Glass e Zoo. Média sobre 30 execuções do algoritmo.

A Figura 47 mostra os valores para a extensão das bases de dados Iris, Glass e Zoo. O valor da extensão para a base Iris assume valores maiores que as demais bases, pois consegue formar componentes conexos, além disso, o tamanho das três classes é igual. Já as bases Glass e Zoo apresentam além da mistura entre os componentes, classes com tamanhos variados, de modo que quando o cálculo da extensão é realizado, ele é ponderado pelo tamanho da maior classe, e com isso, o valor da extensão alcança valores mais baixos.

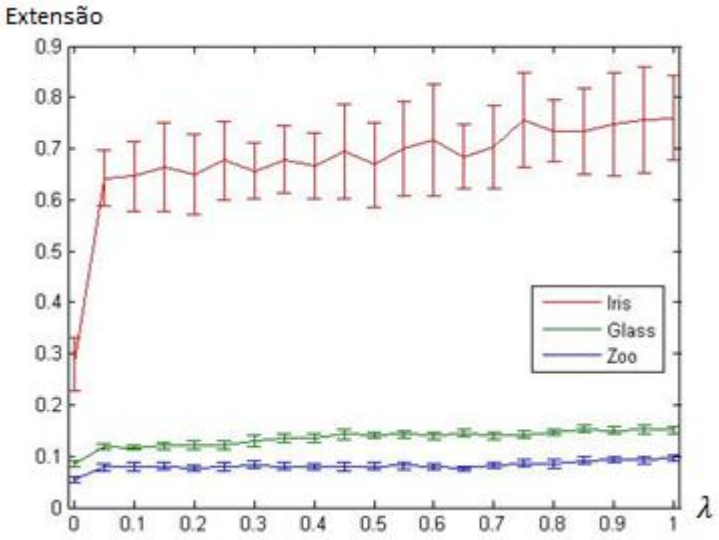


Figura 47: Representação da extensão para as bases de dados Iris, Glass e Zoo. Média sobre 30 execuções do algoritmo.

A Figura 48 mostra os resultados para a energia nas bases Iris, Glass e Zoo quando esta é maximizada em alguns valores de λ. Seu comportamento se assemelha ao das demais bases artificiais, ou seja, decai conforme λ se aproxima de 1 porque passa a dar mais peso para a extensão e essa medida obtém um valor menor que o da pureza para as redes testadas.

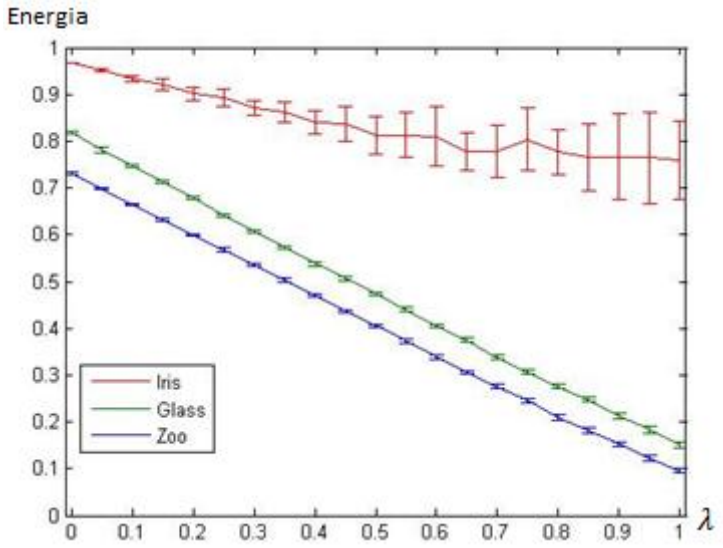


Figura 48: Representação da energia para as bases de dados Iris, Glass e Zoo. Média sobre 30 execuções do algoritmo.

Os resultados obtidos para as bases reais Iris, Glass e Zoo têm um comportamento semelhante ao das bases artificiais analisadas, de modo que bases com mistura menor entre as classes alcançam valores maiores na pureza, extensão e energia, ao contrário de bases com mais mistura. Isso indica que as medidas propostas podem ser usadas para medir a mistura entre classes de dados.

5.4 Discussão do método

Na Seção 5.1 é apresentado um algoritmo para formação de redes a partir de dados proposicionais o qual leva em consideração a similaridade entre os vértices da mesma classe e uma função de energia que pondera medidas de pureza e extensão da rede. A construção de redes a partir de dados é uma fase importante no aprendizado baseado em grafos, de modo que este trabalho pode ser visto como uma contribuição para esta etapa.

Nas Seções 5.2 e 5.3 são mostrados alguns resultados obtidos com a aplicação do método em bases artificiais e reais, variando-se a forma e a quantidade de classes. As redes construídas foram empregadas para caracterizar mistura nas classes de dados. Em todas as bases analisadas nota-se um comportamento semelhante tanto nas redes formadas como nos valores alcançados para pureza, extensão e energia.

Observa-se nas redes formadas que quando λ é igual a 0, a rede final apresenta poucos componentes fortemente conectados quando há pouca mistura entre as classes e um número alto de componentes quando o nível de mistura é maior. Para λ igual a 1, menos componentes são formados nas redes em relação a λ igual a 0.

Os valores para extensão aumentam conforme λ se aproxima de 1, já que esta medida passa a ter mais destaque no cálculo da função da energia. Quando se analisa o comportamento da pureza na rede variando λ, a pureza não apresenta uma queda acentuada conforme λ se aproxima de 1, indicando que mesmo quando um k maior ou um k menor é considerado, os vértices conseguem completar boa parte de suas ligações permanecendo com uma pureza alta. Com isso, redes formadas com um valor maior de λ poderiam ser utilizadas para a classificação quando se tem muita mistura nos dados, pois desse modo, se consideraria redes com um número menor de componentes formados.

Capítulo 6

Conclusões

A fundamentação teórica apresentada neste trabalho tratou das redes complexas e do aprendizado de máquina. Foram apresentados os principais modelos de redes complexas, algumas medidas e aplicações em detecção de comunidades, foi descrito também sobre a detecção de *outliers* e a classificação de dados, as principais técnicas de classificação e as redes *K*-associados.

Uma das propostas apresentadas foi um método de detecção de *outliers* em redes complexas baseado na medida de distância de uma caminhada aleatória e em um índice de dissimilaridade apresentado por Zhou (2003b). Diferente de outras técnicas de detecção de *outlier*, o método proposto possibilita a identificação de diferentes tipos de *outliers*, dependendo da estrutura da rede, os vértices *outliers* podem ser tanto aqueles distantes do centro como os centrais, podem ser *hubs* ou vértices com poucas ligações. De modo geral, são considerados *outliers*, os vértices que possuem um comportamento global diferente da maioria.

Desse modo, podemos concluir que a medida proposta é uma boa estimadora de vértices *outliers* em uma rede, identificando de maneira adequada vértices com uma estrutura diferenciada ou uma função especial na rede. A técnica de detecção de *outlier* se destaca também por reconhecer distintos tipos de *outlier*, diferente de muitas abordagens encontradas na literatura que encontram apenas os vértices mais afastados do centro.

Outra proposta apresentada foi um método para construção de redes baseado nas relações de similaridade entre os vértices da mesma classe e em uma função de energia que leva em consideração medidas de pureza e extensão da rede. O método foi aplicado em alguns

conjuntos de dados artificiais e reais e os resultados foram utilizados para caracterizar mistura entre as classes de dados.

Os resultados obtidos mostraram que conforme a mistura dos dados aumenta, a pureza, a extensão e a energia diminuem, indicando que estas medidas podem ser utilizadas para caracterizar mistura de classes. Além disso, mesmo variando-se a forma dos dados e a quantidade de classes as medidas se comportaram de maneira semelhante.

Observa-se também que conforme a extensão aumenta a pureza não diminui de maneira considerável, indicando que redes formadas com um valor maior de λ poderiam ser utilizadas para a classificação quando se tem muita mistura nos dados, pois desse modo, se consideraria redes com um número menor de componentes formados.

A técnica de caracterização de classes embora apresente resultados preliminares, é um assunto importante e novo. Pode-se considerar que o presente trabalho é uma das primeiras tentativas nesta direção, além de ser uma contribuição para a construção de grafos a partir de dados proposicionais.

6.1 Contribuições

As principais contribuições deste trabalho foram:

- Desenvolvimento de uma medida para identificação de vértices *outliers* em redes complexas, baseada na medida de distância da caminha aleatória de uma partícula Browniana e no índice de dissimilaridade proposto por Zhou (2003b).
- Desenvolvimento de uma técnica para construção de redes a partir de dados proposicionais, baseada nas relações de similaridade entre vértices da mesma classe e nas medidas de pureza e extensões da rede, para caracaterização de mistura entre classes de dados.

As técnicas desenvolvidas originaram as seguintes publicações:

Berton, L.; Huertas, J.; Araújo, B.; L. Zhao (2010) Identifying Singular Nodes in Complex Networks by Using Random Walking Measure. In Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC), vol. 1, p. 2891-2896.

Araújo, B.; Rodrigues, F. A.; Berton, L. Huertas, J. Silva, T. C.; Zhao, L. (2010) Identifying abnormal nodes in protein-protein interaction networks. Dynamics Days South America – International Conference on Chaos and Nonlinear Dynamics – resumo extendido.

Berton, L.; L. Zhao (2011) Caracterização de classes via otimização em redes complexas. VII Encontro Nacional de Inteligência Artificial (ENIA), *in press*.

6.2 Trabalhos futuros

Como trabalho futuro a medida de detecção de *outlier* pode ser utilizada para identificar regiões de simplicidade em redes e similaridade entre vértices. Como muitas vezes as redes complexas possuem regiões de regularidade, onde existem certos padrões de conexões, poderia ser proposto um índice de simplicidade/complexidade e fazer comparações entre redes.

Também é possível utilizar o método de detecção de *outlier* para rotular alguns vértices da rede e a partir disto caracterizar comunidades em uma rede. Considerando que os vértices caracterizados como *outliers* sejam tanto vértices centrais como vértices afastados, ambos são bons candidatos para rotulação a priori, já que são vértices bem representativos de suas comunidades. O método de detecção de *outliers* poderia ser utilizado também no contexto de aprendizado ativo.

Como trabalho futuro o método de construção de redes pode ser aplicado também em mais dados reais, a fim de estudar seu comportamento nestes dados e caracterizá-los. O método pode ser aprimorado, de modo que diferentes medidas podem ser propostas e exploradas, visando extrair outras características de dados com diferentes classes. Podem ser testadas ainda outras medidas de similaridade para composição da rede, além da distância euclidiana. As redes construídas podem ser utilizadas na classificação relacional.

Referências Bibliográficas

Albert, R.; Jeong, H.; Barabasi, A. L. (1999) Diameter of the World Wide Web. *Nature* v.401, p. 130-131.

Albert, R.; Barabási, A. L. (2002) Statistical mechanics of complex networks. *Review of Modern Physics*, v. 74, p. 47-97.

Anton, H.; Rorres, C. (2005) *Elementary Linear Algebra with applications*. Wiley.

Baldi, P.; Brunak, S. (1998) *Bioinformatics: the machine learning approach*. MIT Press.

Barabási, A. L.; Albert, R. (1999) Emergence of scaling in random networks. *Science*, v. 286, p. 509-512.

Barabási, A. L. (2003) *Linked: How Everything Is Connected to Everything Else and What It Means*. Plume.

Barnett, V.; Lewis, T. (1994) *Outliers in Statistical Data*. John Wiley & Sons.

Berton, L.; Huertas, J.; Araújo, B.; L. Zhao (2010) Identifying Singular Nodes in Complex Networks by Using Random Walking Measure. *In Proceedings of the IEEE Congress on Evolutionary Computation* (IEEE CEC), vol. 1, p. 2891-2896.

Blum, A.; Mitchell, T. (1998) Combining labeled and unlabeled data with co-training. *In Proceedings of the eleventh annual conference on Computational learning theory* (COLT'98), p. 92-100.

Boccaletti, S.; Ivanchenko, M.; Latora, V.; Pluchino, A.; Rapisarda, A. (2007) Detecting complex network modularity by dynamical clustering. *Physical Review E*, v. 75, p. 1-4.

Braga, A.; Carvalho, A.; Ludermir, T. (2007) *Redes Neurais Artificiais – Teoria e aplicações*, LTC.

Brandes, U. (2011) A faster algorithm for betweenness centrality. *Journal Mathmatics Sociology*. v. 25, p.163.

Cancho, F.; Solé, R. V. (2003) Optimization in complex networks. Statistical Mechanics of Complex Networks. *Lecture Notes in Physics*, Springer (Berlin), v. 625, p. 114-125.

Chacrabarti, S.; Dom, B.; Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *In SIGMOD´98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, New York, p. 307-318.

Chandola, V.; Banerjee, A.; Kumar, V. (2007) Anomaly Detection - A Survey. *ACM Computing Surveys*, v. 41(3), p. 15.

Chapelle, O.; Shölkopf, B. e Zien, A. (2006) *Semi-supervised Learning. Adaptive Computation and Machine Learning*. Cambridge, MA: The MIT Press.

Chung, F.; Lu, L. (2006) Complex Graphs and Networks. *CBMS Regional Conference Series in Mathematics*, v. 107, p. 264.

Clauset, A.; Newman, M.; Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, v.70 066111.

Cook, D.J.; Holder, L.B. (2000) Graph-based data mining. *IEEE Intelligent Systems*, v.15, p. 32-41.

Dorogovtsev, S. N.; Mendes, J. F. F. (2003) *Evolution of Networks: From Biological Nets to Internet and WWW*. Oxford, p. 280.

Erdös, P.; Rényi, A. (1959) On random graphs. *Publicationes Mathematicae*, v. 6, p. 290-297.

Faloutsos, M.; Faloutsos, P.; Faloutsos, C. (1999) On power law relationships of the internet topology, *Computer Communication Review*, v. 29, p. 251-262.

Freeman, C. L. (1977) Sociometry, v. 40, 35.

Gelfand, A. E.; Smith, A. F. M. (1990) Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, v. 85, p. 398-409.

Geman, S.; Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, v. 6, p. 721-741.

Govindan, R.; Tangmunarunkit, H. (2000) Heuristics for Internet map discovery. *IEEE INFOCOM 2000*, Tel Aviv, Israel, vol. 3, p. 1371-1380.

Grubbs, F. E. (1969) Procedures for detecting outlying observations in samples. *Technometrics*, v. 11(1), p. 1–21.

Hodge, V. J.; Austin, J. (2004) A survey of outlier detection methodologies. *Artificial Intelligence Review*, v. 22, p. 85-126.

Huberman, B.; Adamic, L. (1999) Growth dynamics of the World Wide Web. *Nature*, v. 401, p. 131.

Jeong, H.; Tombor, B.; Albert, R.; Oltvai, Z.; Barabási, A. L. (2000) The large-scale organization of metabolic networks. *Nature*, v. 407, p. 651-655.

Jeong, H.; Mason, S.; Barabási, A.L.; Oltvai, Z. (2001) Lethality e centrality in protein networks. *Nature*, v. 411, p. 41-42.

Lai, Y. C.; Motter, A.; Nishikawa, T.; Park, K.; Zhao, L. (2005) Complex networks: Dynamics and security. *Pramana-Journal of Physics*, v. 64 (4), p. 483–502.

LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. (1989) Backpropagation applied to handwritten zip code recognition. *Neural Computation*, v. 1, p. 541-551.

Lopes, A. A.; Bertini, Jr. J. R.; Motta, R.; Zhao, L. (2009) Classification Based on the Optimal *K*-Associated Network, in *Proceedings of The First International Conference on Complex Sciences: Theory and Applications*, p. 1-11.

Lu, Q.; Getoor, L. (2003). Link-based classification. *In Proceedings of the International Conference on Machine Learning* (ICML). AAAI Press, p. 496-503.

Macskassy, S.; Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, v.8, p. 935-983.

MCulloch, W. S.; Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v.5, p.115-133.

Milgram, S. (1967) *Psychology Today*, v. 1, 60.

Mitchell, T. M. (1997) *Machine learning*. McGraw-Hill Series in Computer Science, McGraw-Hill.

Newman, M. (2001). Who is the best connected scientist? a study of scientific co-authorship networks. *Physical Review E*, v. 64 016132.

Newman, M. (2003) The structure and function of complex networks. *SIAM Review*, v. 45, p. 167-256.

Newman, M. (2004) Fast algorithm for detecting community structure in networks, *Physical Review E*, v. 69 066133.

Newman, M. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, v. *74* 036104.

Newman, M.; Girvan, M. (2004) Finding and evaluating community structure in networks, *Physical Review E*, v. 69 026113.

Noh, J. D.; Rieger, H. (2004) Random Walk on Complex Networks. *Physical Review Letters*, v. 92 118702.

Palla, G.; Der´enyi, I.; Farkas, I. e Vicsek, T. (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, v. 435, p. 814–818.

Quiles, M. G.; Zhao, L.; Alonso, R. L.; Romero, R. A. F. (2008) Particle competition for complex network community detection. *Chaos*, v. 18, p. 1-10.

Raedt, L. D. (2008) *Logical and Relational Learning*. Cognitive Technologies, Berlin Heidelberg: Springer.

Redner, S. (1998) How popular is your paper? An empirical study of the citation distribution. *European Physical Journal B*, v. 4, p. 131-134.

Reichardt, J.; Bornholdt, S. (2004) Detecting fuzzy community structures in complex networks with a Potts model. *Physical Review Letters*, v. 93, p. 1-4.

Rosenblatt, F. (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychology Review*, v. 65, p. 386-408.

Tan, P.; Steinbach, M.; Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley.

Wasserman, S. Faust, K. (1994). *Social Networks Analysis*. Cambridge University Press, Cambridge.

Watts, D. J.; Strogatz, S. H. (1998) Collective dynamics of 'small-world' networks. *Nature*, v. 393, p. 440-442.

Witten, I. A.; Frank, E. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. San Diego: Academic Press.

Woess, W. (2000) *Random walks on infinite graphs and groups*. Cambridge University Press.

Wolpert, D.H., Macready, W.G. (1997) No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, v. 1, p. 67.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, v. 33, p. 452.

Zhou, H. (2003a) Network landscape from a Brownian particle´s perspective. *Physical Review E*, v. 67 041908.

Zhou, H. (2003b) Distance, dissimilarity index, and network community structure. *Physical Review E*, v. 67 061901.

Zhu, X. (2005). *Semi-supervised learning literature survey*. Technical Report 1530, Computer Sciences, Univesity of Wisconsin-Madison.