Guilherme Ferraz de Arruda

Uma abordagem de redes complexas para agrupamento de dados

São Carlos - SP, Brasil

22 de novembro de 2011

Guilherme Ferraz de Arruda

Uma abordagem de redes complexas para agrupamento de dados

Trabalho de conclusão de curso de engenharia elétrica com ênfase em eletônica da Escola de Engenharia de São Carlos da Universidade de São Paulo

Orientador:
Francisco Aparecido Rodrigues

São Carlos - SP, Brasil

22 de novembro de 2011

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

A778a

Arruda, Guilherme Ferraz de.
Uma abordagem de redes complexas para agrupamento de dados. / Guilherme Ferraz de Arruda ; orientador Francisco Aparecido Rodrigues -- São Carlos, 2011.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2011.

1. Agrupamento de dados. 2. Redes complexas. 3. Reconhecimento de padrões. 4. Teoria dos *Grafos*. I. Titulo.

Resumo

Muitos métodos foram desenvolvidos para o agrupamento de dados, como maximização de expectativa, k-médias e algoritmos baseados em teoria dos grafos. Neste último caso, os grafos são geralmente construídos considerando-se a distância euclidiana como medida de similaridade, e particionado usando-se métodos espectrais. No entanto, estes métodos não são precisos quando os clusters não são bem separados. Além disso, não é possível determinar automaticamente o número de clusters. Essas limitações podem ser superadas, considerando-se algoritmos de detecção de comunidades em redes. Este trabalho propõe uma metodologia de agrupamento de dados baseado na teoria de redes complexas. Diferentes métricas são comparadas para quantificar as semelhanças entre objetos e três técnicas de detecção de comunidades são consideradas. O método proposto é aplicado em duas bases de dados de problemas reais e dois conjuntos de dados gerados artificialmente. Ao comparar o método de *clustering* com abordagens tradicionais, verifica-se que a proximidade medidas dado pela exponencial do inverso da distância Chebyshev é a métrica mais adequada para quantificar as semelhanças entre os objetos. Além disso, o método de identificação da comunidade com base na otimização gulosa oferece as menores taxas de erro.

Palavras chave: Agrupamento de dados, Redes complexas, Reconhecimento de padrões, Teoria dos *Grafos*.

Abstract

Many methods have been developed for data clustering, such as k-means, expectation maximization and algorithms based on graph theory. In this latter case, graphs are generally constructed by taking into account the Euclidian distance as a similarity measure, and partitioned using spectral methods. However, these methods are not accurate when the clusters are not well separated. In addition, it is not possible to automatically determine the number of clusters. These limitations can be overcome by taking into account network community identification algorithms. This monograph proposes a methodology for data clustering based on complex networks theory. Different metrics are compared to quantify the similarities between objects and it is taken into account three community finding techniques. This approach is applied two real-world databases and to two sets of artificially generated data. By comparing our method with traditional clustering approaches, we have verified that the proximity measures given by the exponential of the inverse of Chebyshev distance is the most suitable metrics to quantify the similarities between objects. In addition, the community identification method based on the greedy optimization provides the lowest misclassification rates.

Keywords: Data clustering, Complex networks, Pattern recognition, *Graph* Theory.

Dedicatória

Dedico este trabalho aos meus pais Juarez e Ana Sylvia, meus avós, Milton e Jacira (dedicatória póstuma), Maria e Francisco (dedicatória póstuma) ao meu irmão Henrique e minha namorada Livia.

Agradecimentos

Agradeço a todos aqueles que me apoiaram e torceram por mim, em especial meus pais, Juarez e Ana Sylvia. A paciência, compreensão e ajuda de todos, especialmente do meu orientador Francisco Aparecido Rodrigues. Ao meu irmão, Henrique Ferraz de Arruda, pelo apoio e discussões. A e minha namorada, Livia, por tudo.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

1.1 Introdução

O objetivo de qualquer técnica de agrupamento é encontrar uma estrutura de grupos dentro de um conjunto de dados, onde os objetos que pertencem a um mesmo grupo compartilham alguma característica ou propriedade relevante para o domínio do problema, ou seja, de alguma maneira são similares (JAIN; DUBES, 1988). Esta tarefa também é conhecida como *clustering* ou também por classificação não-supervisionada. Diversas técnicas foram propostas na literatura para solução deste tipo de problema, porém este não possui uma solução ótima que possa ser computada em tempo linear. Sendo assim, cada técnica faz uso de uma heurística diferente, fazendo com que cada uma tenha seu viés e encontre grupos diferentes. Não há uma solução única e correta, mas sim uma interpretação diferente dos dados de entrada. Dentro deste contexto deseja-se aplicar a teoria de redes complexas como uma forma de obter informações relativas a estes dados. Na literatura são propostas várias definições de *clusters*, como será visto mais adiante. Cada definição é melhor ajustada a algum algoritmo ou conjunto de dados.

A teoria das redes complexas nasceu da aplicação de medidas desenvolvidas pela teoria dos grafos e conceitos provenientes da mecânica estatística, física não-linear e sistemas complexos. Talvez o primeiro grafo da história, considerando o formalismo matemático, foi sugerido por Leonhard Euler para resolver o famoso problema das *Sete Pontes de Königsberg* (Prússia no século XVIII, atual Kaliningrado, Rússia). Nessa cidade, haviam duas grandes ilhas que, juntas, formavam um complexo que continha sete pontes. Os moradores de Königsberg se perguntavam se seria possível alguém atravessar todas as pontes sem repetir nenhuma. Leonhard Euler ofereceu uma rigorosa prova matemática indicando que tal caminho não existia. Na verdade, ninguém encontrou tal caminho até que uma nova ponte foi construída em 1875. Euler não apenas resolveu o problema de Königsberg, mas sim introduziu um novo ramo da matemática conhecido como teoria dos grafos. Euler modelou o problema transformando as ilhas em arestas e as pontes, conectando as ilhas, em *links*, criando assim, possivelmente o primeiro grafo da história. Isto é mostrado na figura 1.1.

Na década de 1960, a teoria dos grafos ganhou grande impulso com os trabalhos de dois matemáticos Hungaros, Paul Erdös e Alfred Rényi, que propuseram um grafo com estrutura totalmente aleatória. Esta estrutura aleatória é interessante do ponto de vista matemático, porém, não é adequada para representar a estrutura de sistemas reais. Em 1998, os pesquisadores Watts e Strogratz propuseram um modelo que situa-se entre a regularidade e a aleatoriedade. Este modelo é conhecido como *small world*, ou pequeno mundo, em português. Esta topologia mostrou-se eficaz para representar propriedades topológicas de sistemas complexos, tais como a rede neuronal do *Caenorhabditis elegans* e a rede de transmissão de energia dos Estados Uni-

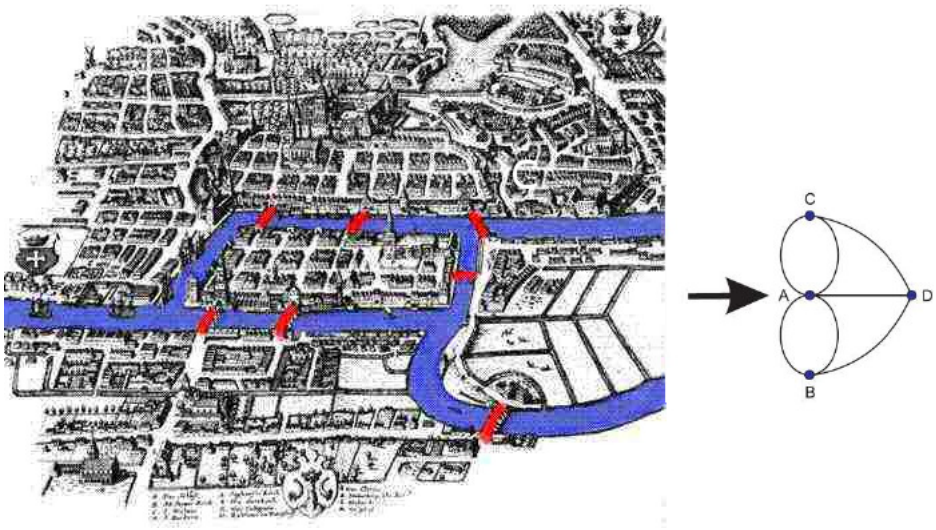

Figura 1.1: Problema das sete pontes de Königsberg. A esquerda têm-se o desenho da cidade com as pontes desenhadas em vermelho. A direita o grafo que modela este problema. Resolver o problema significava encontrar um caminho que passase por todas as pontes, mas sem repetir uma única ponte. Euler resolveu tal problema trocando cada uma das quatro porções de terra por vértices (A até D) e cada ponte por um *link*, obtendo assim um grafo com quatro vértices e sete *links* . Ele provou assim que um caminho cruzando todas as pontes passando apenas uma vez por cada não existia.

dos. Em 1999 os irmão Faloustos (FALOUTSOS; FALOUTSOS; FALOUTSOS, 1999) observaram que o número de conexões (k) da Internet segue uma lei de potência, isto é, $P(k) \sim k^{-\gamma}$. Esta mesma propriedade também foi verificada por Barabási e colaboradores na *World Wide Web* (ALBERT; JEONG; BARABÁSI, 1999). Tais pesquisas despertaram o interesse da comunidade científica na análise da organização de sistemas complexos, gerando milhares de publicações. Nesse trabalho, deseja-se mostrar a aplicabilidade desta teoria dentro do contexto da classificação não-supervisionada.

1.2 Estrutura da monografia

Esta monografia foi dividida em quatro capítulos. Este primeiro capítulo tem a finalidade de contextualizar o leitor dentro dos problemas relacionados ao agrupamento de dados, como também expor a estrutura adotada neste trabalho. No segundo será feita uma introdução e revisão dos métodos tradicionais para agrupamento dos dados, onde são apresentados cada um do métodos que serão utilizados para comparação com a metodologia proposta neste trabalho. São apresentadas as principais características, bem como algoritmos e exemplos para melhor entendimento destes. Além disto, será apresentada uma metodologia para comparação da qualidade

dos grupos fornecidos por algoritmos de agrupamento.

No terceiro capitulo será feita uma breve introdução dos conceitos fundamentais da teoria das redes complexas, já que este é um campo bastante extenso. Serão apresentados métodos para armazenamento computacional deste tipo de estrutura, bem como sua formulação matemática. Será exposta, também, uma metodologia para o reconhecimento de comunidades dentro do contexto de redes complexas. Por fim, será feita a interpretação de um conjunto de dados segundo a estrutura de redes e aplicação dos conceitos apresentados antereriormente.

Finalmente, serão comparados os métodos expostos no segundo capitulo com a metodologia proposta. Esta comparação será feita com bases sintéticas, ou seja, construídas artificialmente de acordo com alguma metodologia, e também com bases de dados reais. Ao final será discutida a aplicabilidade da metodologia proposta dentro do contexto de agrupamento de dados.

2 Agrupamento de dados

2.1 Introdução

Clustering ou agrupamento de dados é uma tarefa de aprendizado não-supervisionado que se refere a identificação de informações relevantes nos dados sem a presença de um elemento externo para guiar o aprendizado. A essência desta modalidade de aprendizado é a identificação de propriedades intrínsecas dos dados de entrada de maneira a construir representação destes. Além de encontrar padrões ou tendência que auxiliem na compreensão destes dados.

Apesar da idéia de *cluster* ser bastante intuitiva não há uma definição formal, única e precisa para este conceito. Em geral cada algoritmo utiliza uma definição que é mais conveniente para determinado tipo de aplicação. Algumas das definições presentes na literatura são (THEODORIDIS; KOUTROUMBAS, 2003): (i) *Cluster* baseado em centro: um *cluster* é um conjunto de pontos tal que qualquer ponto em um dado *cluster* esta mais próximo ao centro deste *cluster* do que qualquer outro *cluster*; (ii) *Cluster* contínuo: é um conjunto de pontos tal que qualquer ponto em um dado *cluster* é mais similar a um ou mais pontos nesse *cluster* do que a qualquer ponto que não pertence a ele; (iii) *Cluster* baseado em densidade: um *cluster* é uma região densa de pontos, separada de outras regiões de alta densidade por regiões de baixa densidade; (iv) *Cluster* baseado em similaridade: é um conjunto de pontos similares, enquanto pontos de clusters diferentes são menos similares.

Existem várias técnicas para pré-processamento dos dados em agrupamento de dados (THEODORIDIS; KOUTROUMBAS, 2003; BISHOP, 2006), como análise de componentes principais, PCA do inglês, *Principal Component Analysis*, *Kernel PCA*, *Probabilistic PCA*, dentre muitas outras. Estas não serão utilizadas neste trabalho, pois aqui deseja-se enfatizar o método em si e não o efeito da dimensionalidade. Porém será utilizada a técnica de normalização dos dados, dado pela expressão a seguir:

$$y_f = \frac{x_f - \overline{x_f}}{\sigma_{x_f}} \tag{2.1}$$

onde $\overline{x_f}$ e σ_{x_f} são, respectivamente, é a média o desvio padrão do atributo. Esta técnica elimina efeitos de escala entre os diversos atributos. É possível que, em determinado conjunto de dados, haja atributos que variem em faixas muito distintas, como por exemplo, um atributo varie na faixa $[0,1]$ e outro na faixa $[0,1000]$. Com isso, o modelo induzido pelo algoritmo de agrupamento será influenciado pela escala dos atributos e não pela estrutura dos dados. Ao aplicar a normalização dos dados este problema é eliminado, pois os atributos ficarão normalmente distribuídos com média igual a zero e variância unitária. Além disso, o uso desta técnica é mais aconselhado que uma simples mudança de escala, pois tende a tratar melhor os *outliers* (THE-

ODORIDIS; KOUTROUMBAS, 2003; BISHOP, 2006).

Nas seções posteriores será feita uma revisão e apresentação dos principais métodos de agrupamento de dados, bem como de uma metodologia para avaliação da qualidade dos *clusters* fornecidos por qualquer método de agrupamento. Os métodos k-médias (seção 2.2), cobweb (seção 2.4), *farthest first* (seção 2.5) e *expectation maximization* (seção 2.6) serão utilizados para comparação com o método proposto. Quanto ao método hierárquico (seção 2.3), este não será utilizado durante a comparação com a metodologia proposta, mas é uma das principais abordagens de agrupamento e constitui base para melhor entendimento de outros métodos, como por exemplo o *cobweb*, bem como compreensão do método de reconhecimento de comunidades guloso, que será apresentado na seção 3.2.1.

2.2 k-Médias

K-médias é um dos algoritmos mais populares para agrupamento de dados. Seja um conjunto de dados $X = \{\vec{x}_1, ..., \vec{x}_N\}$ com N observações, onde cada vetor $\vec{x}_i = (x_{i1}, x_{i2}, ..., x_{im})$ deste conjunto é formado por m atributos. Este algoritmo consiste na minimização do erro quadrático para um número de grupos, k, fixo, isto é,

$$E = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, \bar{x}^{(j)})^2, \tag{2.2}$$

onde $d(x_i, \bar{x}^{(j)})$ é uma distância e $\bar{x}^{(j)}$ é o centróide do cluster C_j, definido pela média dos elemento pertencentes à este grupo. Sendo assim, o erro quadrático é a soma das variações dentro dos *clusters*. A minimização desta função garante a propriedade de compactação dos *clusters*, que é equivalente à maximizar a variação entre *clusters* (JAIN; DUBES, 1988). A distância usualmente utilizada para esta tarefa é a distância euclidiana, o que resulta em grupos esféricos em torno dos k centróides. Porém é possível utilizar-se outras métricas e alterar a forma dos agrupamentos. Uma possibilidade é a utilização da distância de Mahalanobis, que pode ser empregada para encontrar grupos hiperelipsoidais. Outra possibilidade é a utilização da distância de Minkowski, que será apresentada na seção 3.3.

O algoritmo k-médias tem início distribuindo um conjunto de k centróides para os *clusters*. Esta tarefa pode ser realizada de diversas formas, a mais comum é a escolha aleatória de k objetos do conjunto de dados. Em seguida, cada ponto do conjunto de dados é associado ao *cluster* com o centróide mais próximo. Após esta etapa, os centróides são recalculados como sendo a média dos pontos presentes no *cluster*. Este processo é repetido até que os centróides não sejam

mais alterados ou seja excedido um número máximo de iterações definido previamente. Este procedimento é ilustrado em pseudo código abaixo e exemplificado na figura 2.1.

Algorithm 1 Algoritmo k-médias básico

Require: Conjunto de dados X_{nxd} e Número de *clusters*.
Ensure: Uma partição de X em k *clusters*.
  repeat
    for all objeto em $x_i \in X$ e *cluster* $C_j, j = 1,2,...k$ **do**
      Calcule a distância entre x_i e o centróide do *cluster* $\overline{x}^{(j)} : d(x_i, \overline{x}^{(j)})$, utilizando a métrica de distância definida.
    end for
    for all objeto em x_i **do**
      Associar x_i ao centróide mais próximo.
    end for
    for all *cluster* $C_j, j = 1,2,...k$ **do**
      Recalcular o centróide.
    end for
  until não houver alterações na associação dos grupos

A grande vantagem do método k-médias é que ele possui complexidade de tempo *O(n)*, uma vez que o número de iterações é tipicamente pequeno e $k \gg n$.

Como pontos negativos, deve-se destacar que este algoritmo é sensível à escolha inicial dos centróides, além de convergir para ótimos locais.

2.3 Agrupamento hierárquico

Esta classe de algoritmos gera, a partir de uma matriz de similaridade, uma sequência de partições aninhadas. Estes podem ser divididos em duas abordagens, a aglomerativa e a divisiva, como demonstrado na figura 2.2. A primeira abordagem inicia-se a partir de um conjunto de *n clusters* com um único objeto cada agrupando sucessivamente estes objetos até obter apenas um único grupo. A segunda inicia-se a partir de um único grupo que contém todos os objetos e realiza sucessivas divisões, até que todos os grupos possuam apenas um objeto.

A técnica de agrupamento hierárquico é bastante flexível em relação ao nível de granularidade, é fácil a utilização de qualquer forma de similaridade (ou dissimilaridade), além da possibilidade de usar qualquer tipo de atributo. Como aspectos negativos deve-se destacar que este algoritmo é guloso e após realizar uma escolha, este não volta atrás, podendo realizar escolhas sub-ótimas, além disto, o número de *clusters* é uma informação necessária para se particionar um conjunto de dados.

A escolha de um métrica influencia a forma dos clusters gerados. Algumas possibilidades

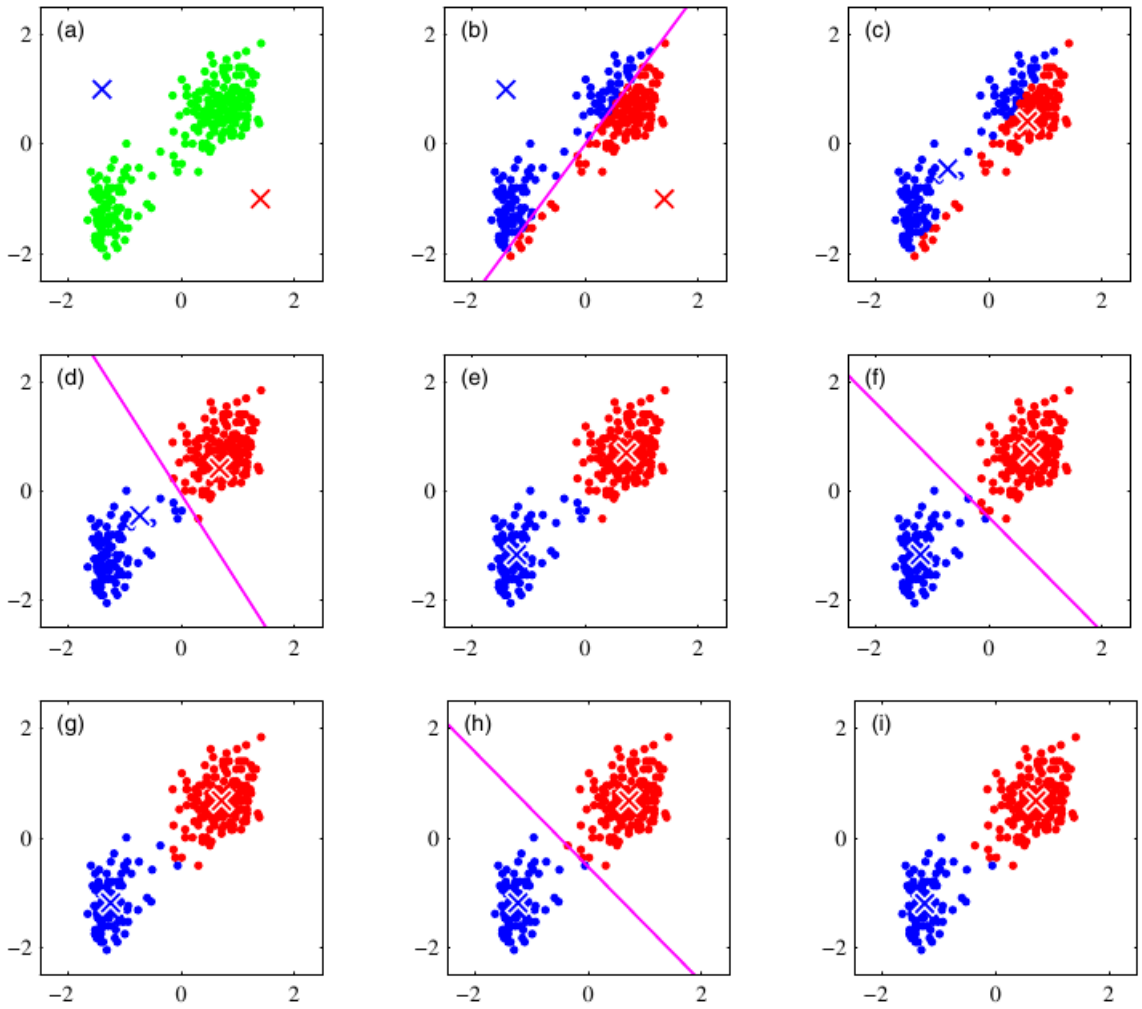

Figura 2.1: Ilustração do funcionamento do método k-Médias utilizando uma versão re-escalada da base de dados *Old Faithful*. (a) Denota os pontos da bases de dados em um espaço bidimensional euclidiano. Escolha inicial dos centróides são mostrados como duas cruzes, uma azul e outra vermelha. (b) Cada pondo do conjunto de dados é associado a um dos centróides de acordo com a proximidade. Isto é equivalente à classificar os pontos de acordo com a linha em magenta. (c) Nesta etapa cada centróide é recalculado como sendo a média dos pontos do seu *cluster*. (d) - (i) Mostra a execução sucessiva destes passos até a convergência final do algoritmo. Figura retirada de (BISHOP, 2006).

são mostradas na seção 3.3, como por exemplo a distância Minkwoski, a partir da qual é possível obter as distâncias euclidiana e Manhattan. Esta classe de algoritmos não possui uma função objetivo global e são baseados em decisões locais. Além da métrica do espaço é preciso definir a distância entre *clusters*, também conhecidas por métricas de ligação (*linkage metrics*). As métricas mais comuns são:

- *Complete linkage*: $\max\{d(a,b) : a \in A, b \in B\}$

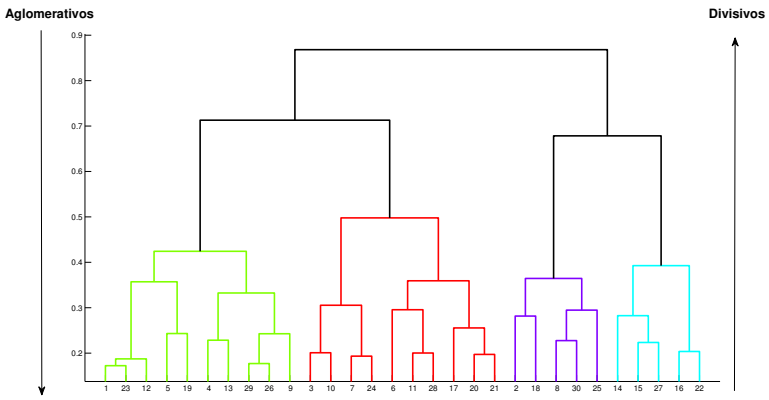

Figura 2.2: Exemplo de dendrograma obtido por um método de agrupamento hierárquico. As setas indicam o funcionamento de algoritmos aglomerativos e divisivos.

- *Single-linkage*: $\min\{d(a,b) : a \in A, b \in B\}$
- *Average linkage*: $\frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$

A métrica utilizada pelo algoritmo com ligação simples (*Single-linkage*) é indicada para manipular formas não elípticas, mas é muito sensível a ruídos e *outliers*. Em geral favorece clusters finos e alongados. Já o algoritmo com ligação máxima (*Complete linkage*), é menos suscetível à ruídos e *outliers*, mas tem a tendência de quebrar grupos grandes e apresenta uma deficiência com formas convexas. Em geral favorece a formação de grupos esféricos.

Abaixo é descrito um algoritmo básico, em pseudo código, para agrupamento hierárquico aglomerativo. Com algumas alterações é possível obter um algoritmo divisivo.

Algorithm 2 Algoritmo hierárquico aglomerativo

Require: Uma matriz de dissimilaridade entre os pares de objetos S_{NxN}.
Ensure: Uma hierarquia de partição.
  Alocar cada objeto a um *cluster*.
  while Há *clusters* para agrupar **do**
    Calcular uma matriz de distância entre os pares de *clusters* disponíveis, utilizando uma métrica de integração (*linkage metrics*).
    Combinar o par de *clusters* C_i e C_j mais próximos, gerando um único *cluster* C_{ij}.
  end while

Esta classe de algoritmos, em geral, não lida bem com ruídos e *outliers* e depende da ordem de entrada dos dados, pois utiliza uma estratégia gulosa. Porém esses métodos não requerem informações prévias sobre o número de grupos, uma vez que esta informação pode ser fornecida

ao final do processo, com o dendrograma em mãos. Além disto, seu resultado corresponde a uma taxonomia dos dados e pode fornecer informações importantes sobre o conjunto de dados.

2.4 *CobWeb*

CobWeb é um sistema de agrupamento conceitual, que organiza os dados visando a maximização da habilidade de inferência. É também um algoritmo incremental e de baixo custo computacional, além de ser bastante flexível , podem ser aplicado em vários domínios (FISHER, 1987).

Esta técnica constitui um sistema incremental para agrupamento hierárquico conceitual. Ele utiliza a heurística de busca *hill-climbing* no espaço de possíveis sistemas utilizando operadores que permitem percorrer este espaço de maneira bidirecional. Assim, este não é puramente aglomerativo ou divisivo, como os algoritmos que foram apresentados na seção 2.3, mas utiliza, em sua construção, operadores aglomerativos (*merging*) e divisivos (*splitting*) que serão apresentados a seguir.

Estratégia de busca: *Hill-climbing*

Hill-climbing é uma técnica de otimização matemática que pertence à família de busca local. É um algoritmo iterativo que se inicia com uma solução arbitrária para um problema, tentando posteriormente encontrar uma solução melhor através da mudança progressiva de um único elemento da solução. Se a alteração produz uma solução melhor, uma mudança incremental é feita para a nova solução, repetindo até que não haja mais melhorias a serem feitas.

Esse algoritmo possibilita encontrar um ótimo local, ou seja, uma solução que não pode ser melhorada, considerando-se uma configuração de vizinhos. Logo, não é garantida a sua convergência para o ótimo global dentro do espaço de busca. Uma modificação possível para este algoritmo é o *simulated annealing*, que será apresentado no contexto de reconhecimento de comunidades em redes complexas, na seção 3.2.2.

Heuristica para busca: *category utility*

A heurística de busca utilizada pelo *CobWeb* é chamada *category utility*, proposta por (CORTER; GLUCK, 1985) como uma métrica para prever o nível básico em hierarquias de classificação humana. Nesse caso, os níveis básicos são retornados mais rapidamente que outros mais gerais ou específicos durante o reconhecimento de objetos. O exemplo sugerido em (FISHER, 1987) é que, ao pensar em uma estrutura hierárquica de animais, categorias como pássaros são re-

tornadas mais rapidamente que outras mais gerais como animais (topo da estrutura) ou mesmo beija-flor, que é um tipo de pássaro. Mais especificamente, categorias de nível básico são, por hipótese, aquelas onde o número de habilidades relacionadas a inferência é maximizado em humanos (MERVIS; ROSCH, 1981).

Identificar os conceitos que são utilizados por seres humanos em processos cognitivos é o principio de critérios de avaliação em sistemas de inteligência artificial. A utilidade de uma determinada categoria pode ser vista como uma função que recompensa virtudes realizadas pelo agrupamento. Tais funções podem retornar por exemplo, a similaridade entre elementos de um mesmo grupo e a dissimilaridade entre elementos de grupos distintos. Em particular, a utilidade de uma categoria é dada por um balanço entre similaridade dentro da classe e dissimilaridade entre as classes de objetos, sendo que os objetos são descritos por pares atributos - valor. A similaridade entre os elementos de uma mesma classe são reflexo da probabilidade condicional da forma $P(A_i = V_{ij}|C_k)$, onde $A_i = V_{ij}$ é um par atributo - valor e C_k é a classe. Quanto maior for a probabilidade, maior será a proporção de membros da classe compartilhando o valor e, assim, mais fácil de se prever o valor dos membros da classe. Já a similaridade entre elementos de classes distintas é uma função da probabilidade $P(C_k|A_i = V_{ij})$. Quanto maior for o valor desta probabilidade, menos objetos em classes distintas compartilham tal valor, assim, mais preditivo ele é em relação a classe.

Estas probabilidades são relativas a cada objeto, mas elas podem ser combinadas, fornecendo uma medida de qualidade da partição, onde uma partição é um conjunto de classes mutuamente exclusivos, ou seja, $\{C_1, C_2, ..., C_n\}$, sendo $C_l \cap C_m = \emptyset, \forall l, m$. Assim, tem-se:

$$\sum_{k=1}^{N} \sum_{i} \sum_{j} P(A_i = V_{ij}) P(C_k|A_i = V_{ij}) P(A_i = V_{ij}|C_k), \tag{2.3}$$

que é um balanço entre similaridades entre elementos de mesma classe (intra classe) e dissimilaridade entre elementos de classes distintas (inter classe) somados em todas as classes, k, todos os atributos, i, e valores, j. Sendo que a probabilidade $P(A_i = V_{ij})$ pondera a importância de valores individuais.

A função 2.3 é um balanço entre similaridade intra classes e dissimilaridades inter classes, porém, ela também recompensa o potencial de inferência de uma classe de um objeto. Mais precisamente, para todo i, j, k, $P(A_i = V_{ij})P(C_k|A_i = V_{ij}) = P(C_k)P(A_i = V_{ij}|C_k)$, pela regra de Bayes e substituindo em 2.3:

$$\sum_{k=1}^{N} P(C_k) \sum_{i} \sum_{j} P(A_i = V_{ij}|C_k)^2, \tag{2.4}$$

o termo $\sum_i \sum_j P(A_i = V_{ij}|C_k)^2$ é o número esperado de atributos - valores que podem ser corretamente descoberto por um membro arbitrário da classe C_k.

Por fim, Gluck e Corter definem a utilidade da categoria como o aumento no número esperado do valor dos atributos que podem ser corretamente descobertos ($P(C_k)\sum_i \sum_j P(A_i = V_{ij}|C_k)^2$), dada uma partição $\{C_1, C_2, ..., C_n\}$, sobre o número esperado de descobertas corretas sem conhecimento ($\sum_i \sum_j P(A_i = V_{ij})^2$). Formalmente:

$$CU(C_1, C_2, ..., C_n) = \frac{\sum_{k=1}^{N} P(C_k)[P(C_k)\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2]}{N}, \quad (2.5)$$

o denominador N é o número de categorias em uma partição. A extração da média de categorias permite a comparação de diferentes tamanhos de partição.

Representação de conceitos

A base de qualquer algoritmo de classificação é a representação individual de conceitos. A probabilidade de um atributo possuir um valor é computada a partir de dois inteiros. O exemplo exposto em (FISHER, 1987) é: o conceito para a classe de pássaros tem uma entrada $P(voar|passaro)$, isto é calculado por $\frac{\sharp\text{Número de vezes que um pássaro foi observado voando}}{\sharp\text{Número de vezes que um pássaro foi observado}}$. Ambos contadores são armazenados no nó "Pássaros".

No *CobWeb*, um conceito probabilístico rotula cada nó da árvore de classificação e sumariza o objeto classificado no nó.

Operações e controle

CobWeb incorpora novos objetos de maneira incremental em uma árvore de classificação, onde cada nó é um conceito probabilístico que representa uma classe. A incorporação de um novo objeto é um processo de classificação do objeto, descendo pela árvore pelo caminho adequado, atualizando-o e realizando algumas operações em cada nível. Estas operações são:

- Classificação do objeto em relação à uma classe já existente;
- Criação de uma nova classe;
- Combinar duas classes em apenas uma;
- Dividir uma classe em outras classes.

Estes operadores são utilizados para realização da busca, por meio da heurística *Hill-climbing*.

Provavelmente a maneira mais intuitiva de atualizar um conjunto de classes é simplesmente colocar o novo objeto em uma das classes já existentes. Para realizar esta tarefa é necessário determinar qual classe melhor recebe o novo objeto. O algoritmo *CobWeb* tenta colocar o objeto em cada categoria existente. O nó que obtiver a melhor partição é o mais adequado para recebê-lo.

Além de colocar os objetos em uma classe existente é possível criar uma nova classe. A qualidade do particionamento resultante de alocação do objeto na melhor categoria é comparado com a qualidade de se criar uma nova classe contendo este objeto. Assim o algoritmo define automaticamente o número de classes. Este número não é um parâmetro, mas aparece naturalmente.

As duas operações já descritas, colocar em uma classe já existente e criar uma nova, são muito eficientes na grande maioria dos casos, porém são extremamente sensíveis a ordem inicial dos dados. Para reduzir este efeito são utilizados outros dois operadores: (i) *Merging*, (ii) *Splitting*. A primeira envolve a criação de um novo nó a partir de outros dois. Somam-se os atributos dos nós que serão unidos em um novo nó e os dois nós originais serão colocados como filhos deste novo nó. Já a segunda envolve a remoção de um nó e a promoção de seus filhos. Estas operações são inversas e permitem ao *CobWeb* fazer uma busca bidirecional no espaço de possíveis hierarquias. Além disso elas diminuem a sensibilidade do algoritmo à ordem inicial dos dados de entrada.

Estrutura de controle

O algoritmo a seguir resume a estratégia de controle utilizada pelo *CobWeb*:

É importante observar que a árvore é construída de maneira incremental e recursiva. Também, as operações de *merging* e *splitting* dessensibilizam o sistema em relação à ordem dos dados de entrada, enquanto a alternativa de busca empregada, *hill-climbing*, é sensível à esta. Assim sendo, os pontos negativos de cada etapa do algoritmo tende a se contrabalançar.

2.5 *Farthest-first*

Farthest-first é uma implementação do algoritmo *k-center*, que consiste em um problema combinatório em que, dado N vértices com distâncias especificas, deseja-se determinar k centróides

Algorithm 3 COBWEB(Objeto, Raiz)

Require: Um objeto, Raiz da árvore de classificação.
Atualiza contador da raiz.
if Raiz é uma folha? **then**
return Folha expandida para acomodar um novo objeto.
else
Encontre qual filho da raiz melhor abriga o novo objeto e realize uma das operações a seguir:
(a) Considere criar uma nova classe e faça-o se apropriado.
(b) Considere a operação *merging*, faça-o se apropriado e chame COBWEB(Objeto, Nó unido (*Merged Node*)).
(b) Considere a operação *splitting*, faça-o se apropriado e chame COBWEB(Objeto, Raiz).
if Caso nenhuma das operações acima (a,b ou c) for realizada **then**
COBWEB(Objeto, Melhor filho da raiz)
end if
end if

em diferentes vértices que minimizem a distância entre o vértice mais distante e o centróide. Na teoria dos grafos, isto significa encontrar k vértices onde a máxima distância à qualquer ponto é a um destes k vértices é mínima. Os vértices devem estar em um espaço métrico, ou seja, devem respeitar a desigualdade triangular (HOCHBAUM; SHMOYS, 1985).

Seja um conjunto de dados X, particionado em K grupos, $\{C_1, C_2, ..., C_K\}$, sendo que o tamanho de cada cluster C_k será o maior valor D em que todos os pontos em C_i são: (i) com distância D de qualquer outro ponto; ou (ii) com distância $\frac{D}{2}$ de algum ponto chamado centróide do *cluster*. Assim, chama-se D_k o tamanho do *cluster* C_k. Por fim, o tamanho da partição S é:

$$D = \max_{k=1,2,...,K} D_k. \tag{2.6}$$

Assim o objetivo deste método é, dado um valor de K, obter $\min_S D(S)$. A formulação matemática deste método é dada por:

$$\min_S \max_{k=1,2,...,K} \max_{i,j:x_i,x_j \in C_k} L(X_i, x_j) \tag{2.7}$$

onde $L(X_i, x_j)$ denota uma distância entre um par de objetos.

Este problema exige a aplicação de alguma técnica de otimização, pois trata-se de um problema NP-hard (HOCHBAUM; SHMOYS, 1985). O algoritmo guloso foi utilizado para solução deste. Ele é mostrado em pseudo código abaixo:

O algoritmo guloso garante uma aproximação de fator 2 como a maior medida de distância,

Algorithm 4 K-Center

H denota o conjunto de objetos representativos do *cluster*, $\{h_1,...,h_k\} \subset S$.
Seja $cluster(x_i)$ a identificação do *cluster* ao qual o elemento $x_i \in S$ pertence.
Seja $dist(x_i)$ a distância entre x_i e o objeto representativo mais próximo do cluster: $dist(x_i) = min_{h_j \in H} L(x_i, h_j)$
Selecione aleatoriamente um objeto x_j de S, seja $h_1 = x_j, H = h_1$.
for $j = 1$ até n **do**
  $dist(x_j) = L(x_j, h_1)$
  $cluster(x_j) = 1$
end for
for $i = 2$ até K **do**
  $D = \max_{x_j} : x_j \in S\, Hdist(x_j)$
  Escolha $h_i \in S\, Hs.t.dist(hi) = D$
  $H = H \in \{h_i\}$
  for $j = 1$ até n **do**
    if $L(x_j, , h_i) \leq dist(x_j j)$ **then**
      $dist(xj) = L(x_j, h_i)$
      $cluster(x_j) = i$
    end if
  end for
end for

L, satisfazendo a desigualdade triangular, isto é:

$$D^* = \min_S \max_{k=1,2,...,K} \max_{i,j:x_i,x_j \in C_k} L(X_i, x_j) \tag{2.8}$$

o algoritmo guloso garantirá:

$$D \leq 2D^* \tag{2.9}$$

Esta relação aparece se o tamanho do *cluster* é definido no sentido de *cluster* centralizado. As provas e deduções podem ser encontradas em (HOCHBAUM; SHMOYS, 1985).

Como propriedades deste algoritmo deve-se enfatizar que este apresenta complexidade de tempo $O(kN)$ e o resultado é uma aproximação com fator 2 se o tamanho da partição S é definido no sentido de *cluster* centralizado.

2.6 *Expectation Maximization*

O método de agrupamento de dados *expectation maximization* (EM) surgiu da unificação de diversos trabalhos apresentados por Dempster et al (DEMPSTER; LAIRD; RUBIN, 1977). De maneira geral, se uma variável foi observada algumas vezes e outras não, é possível utilizar os casos observados para aprender e predizer os valores não observados. O algoritmo EM realiza

esta tarefa, mas também pode ser utilizado para variáveis cujos valores nunca foram observados, sempre e quando seja conhecida a forma geral da distribuição de probabilidade das variáveis.

Em resumo o algoritmo EM é definido em dois passos: (i) **Passo E**: Encontra-se os valores esperados das estatísticas suficientes para os dados completos Y, dado os dados incompletos Z e as estimativas dos parâmetros; (ii) **Passo M**: Utiliza-se estas estatísticas suficientes para fazer uma estimativa de máxima verossimilhança. Considere que $X = x_1, ..., x_m$ são os dados observados independentemente e $Z = z_1, ..., z_m$ os dados não observados nestas instâncias,e seja $Y = X \cup Z$. Z pode ser tratado como uma variável aleatória cuja distribuição de probabilidades depende do conjunto de parâmetros desconhecidos θ e dos dados observados X. Analogamente, Y é uma variável aleatória, já que esta é definida em função da variável aleatória Z. Para descrever a forma geral do algoritmo EM, denota-se a hipótese dos parâmetros atuais, θ por h e a hipótese revisada, que é estimada a cada iteração do algoritmo, por h'

O algoritmo EM consiste na busca pela hipótese h' de maximização da verossimilhança, isto é, que maximize $E[\ln(P(Y|h'))]$. Sendo que este valor esperado é calculado sobre a distribuição de probabilidades de Y, que é determinada pelos parâmetros desconhecidos θ. Sendo que os dados Y são uma combinação dos dados observados X e não observados Z, obtêm-se o valor de $E[\ln(P(Y|h'))]$ sobre a distribuição de probabilidades de Y, que é determinada pelos valores conhecidos X mais a distribuição de probabilidades de Z. Em geral a distribuição de probabilidades de Y não é conhecida, pois ela é determinada pelos parâmetros θ que se deseja estimar. Entretanto o algoritmo EM usa sua hipótese atual h no lugar do parâmetro θ atual para determinar a distribuição de probabilidades de Y. Assim, considere uma função $Q(h'|h)$ que dá $E[\ln(P(Y|h'))]$ como função de h', pela suposição que $\theta = h$ e dada a porção dos dados observados X dos dados Y, tem-se:

$$Q(h'|h) = E[\ln(P(Y|h')|h, X)] \tag{2.10}$$

Assim, formalmente, o algoritmo EM repete os dois passos seguintes até a convergência: **Passo E (*Expectation (E)*)**: Calcula $Q(h'|h)$ utilizando a hipótese atual h e os dados observados X para estimar a distribuição de probabilidades de Y, equação 2.10. **Passo M (*Maximization (M)*)**: troca-se a hipótese h pela h' que maximize a função Q:

$$h = \arg\max_{h'} Q(h'|h) \tag{2.11}$$

Quando a função Q tem um único máximo o algoritmo convergirá para ele, caso contrário ele poderá convergir para máximos locais. O algoritmo EM possui um forte embasamento estatístico, o que justifica o destaque que este tem ganhado atualmente.

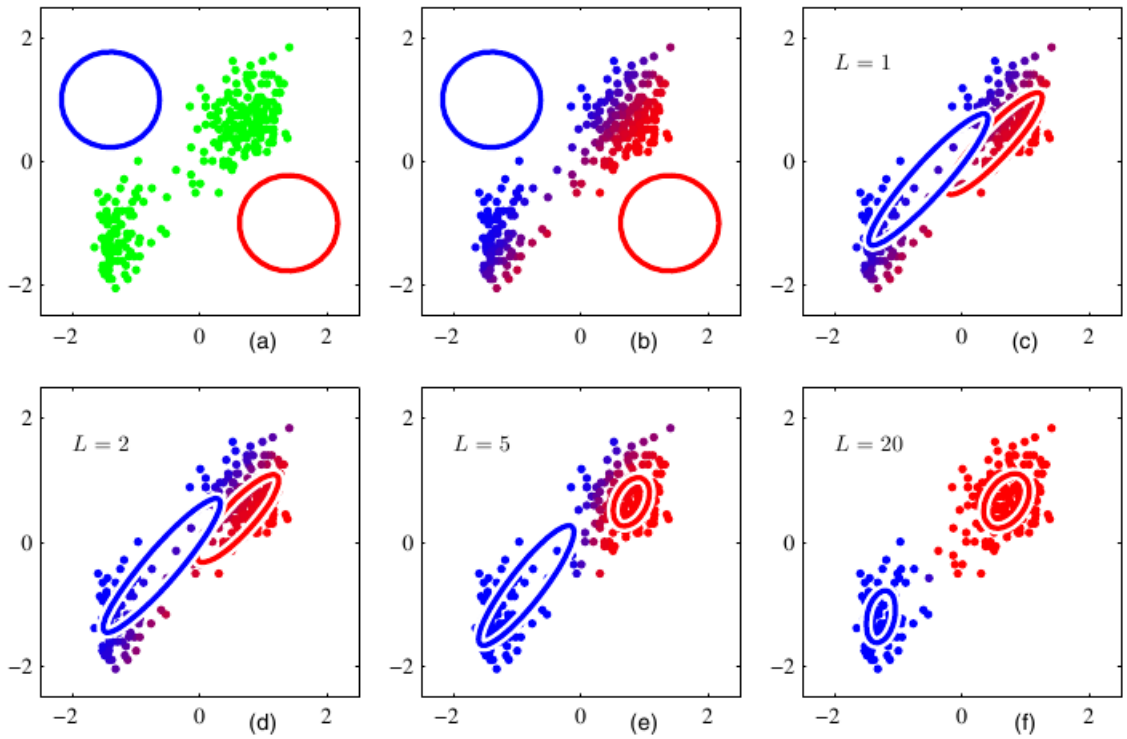

Figura 2.3: Ilustração do algoritmo *Expectation Maximization* usando a base de dados *Old Faithful*, a mesma utilizada para ilustrar o método k-Médias, ver figura 2.1. Figura retirada de (BISHOP, 2006).

Uma ilustração desta técnica é feita na figura 2.3. Além disso a figura 2.1 também pode ser interpretada como aplicação deste método. Nesta o item (b) representa a etapa E e o item (c) a etapa M. As etapas (d) até (i) são repetições sucessivas das etapas E e M.

Originalmente, este algoritmo necessita que seja informado o número de grupos presentes na base de dados. Porém para contornar esta deficiência, pode realizar a validação cruzada com o método descrito acima (esta não será descrita aqui, pois foge do escopo do trabalho, para mais informações veja a referência (TAN et al., 2006)) para estimar o valor da log-verossimilhança, $E[\ln(P(Y|h')|h,X)]$, e aumentar o número de grupos gradualmente enquanto seu valor estiver crescendo. Buscando, assim o maior valor de verossimilhança.

2.7 Validação de agrupamentos

Uma característica natural dos algoritmos de agrupamento de dados é a imposição de uma estrutura de grupos aos dados, mesmo que os dados não possuam esta estrutura. Nesta seção serão apresentadas duas alternativas para verificar a solidez da estrutura proposta. Um

índice muito comum para esta tarefa é o conhecido índice de Jaccard (THEODORIDIS; KOUTROUMBAS, 2003), que é definido pela equação

$$J(C,K) = \frac{a}{a+b+c}, \tag{2.12}$$

onde a denota o número de pares de amostras que possuem a mesma classe em C e são agrupadas no mesmo grupo em K, b é o número de pares com a mesma classe, porém são atribuídas a grupos diferentes e c é o número de pares no mesmo grupo, mas com classes diferentes. Este índice produz resultados no intervalo $[0,1]$, onde o valor 1.0 indica que C e K são idênticos.

Uma alternativa mais natural é considerar o erro. Porém nesta abordagem devem-se considerar casos onde o número de grupos obtido pelo método de agrupamento pode ser diferente do número de classes presentes no conjunto de dados. Assim, considera-se que o *cluster* que mais possui elementos de uma determinada classe é considerado como acerto e os demais *clusters* que também representarem esta mesma classe serão considerados como erros. Desta maneira apenas um cluster representará apenas uma classe. Esta metodologia é mais intuitiva, porém o índice de Jaccard é mais adequado para análise da solidez de agrupamentos. De fato, esta abordagem por meio do erro constitui uma generalização da matriz de confusão presente em problemas de classificação supervisionada.

3 Agrupamento utilizando redes complexas

3.1 Representação de redes complexas

Redes complexas são formadas por um conjunto de vértices (também chamados nós), $\mathscr{N} = \{n_1, n_2, ..., n_N\}$, que são conectados por um conjunto de arestas (*links*), $\mathscr{L} = \{l_1, l_2, ..., l_M\}$, devido a algum tipo de interação. De maneira geral, redes complexas são grafos com estrutura altamente irregular (BOCCALETTI et al., 2006). Computacionalmente, pode-se representar uma rede como uma lista de conexões ou como uma matriz de adjacências. No caso de listas, as conexões são armazenadas como pares (i, j, w_{ij}), onde i e j são os vértices da aresta, e w_{ij} é o respectivo peso desta aresta. Observe que este último parâmetro pode ser omitido em uma rede não-ponderada. Na representação matricial (ver Figura 3.1), a rede é representada por uma matriz A, de dimensão $N \times N$, onde N é o número de vértices, cujos elementos a_{ij} são iguais a 1 se existir uma conexão entre os vértices i e j e iguais a zero, caso contrário. Em redes não-dirigidas, a matriz de adjacências é simétrica, *i.e.* $a_{ij} = a_{ji}$, enquanto que em redes dirigidas, geralmente $a_{ij} \neq a_{ji}$. No caso de redes ponderadas, temos uma matriz de pesos, W, cujo elemento w_{ij} representa o peso da ligação entre os vértices i e j (Ver figura 3.2). Este será o tipo de rede utilizado para representar os dados a serem agrupados, de maneira semelhante à matriz de similaridade utilizada pelos algoritmos hierárquicos (seção 2.3). Em redes complexas, geralmente não ocorre a presença de mais de uma aresta entre dois vértices, nem apresenta de auto-conexões, isto é, uma aresta que liga um vértice a ele mesmo.

Dado um conjunto de dados com N elementos, onde cada elemento é representado por um vetor de atributos, $\overrightarrow{x} = (x_1, x_2, ..., x_N)$ pode ser interpretado como uma rede totalmente conectada com N nós, sendo que cada elemento do conjunto de dados é mapeado por um vértice desta rede, e a intensidade de interação entre cada par de nós da rede é dado por uma medida de similaridade.

Porém esta matriz não é esparsa, agregando um maior custo computacional, podendo, até, inviabilizar a aplicação do método proposto para grandes bases de dados. Uma alternativa a este problema é a aplicação de um limiar, considerando apenas as conexões mais importantes, porém isto deve ser feito de maneira que a rede seja conexa, ou seja, sempre haverá um caminho que conecta dois pares de vértices. A maneira utilizada para realizar esta operação de limiarização foi a seleção de um determinado percentual de conexões para cada nó, levando em consideração apenas informações locais e não considerando informações globais da estrutura da rede.

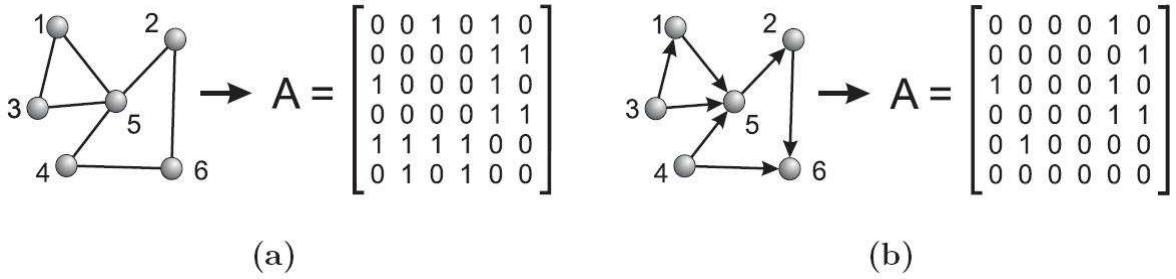

Figura 3.1: Redes complexas podem ser representadas por matrizes de adjacência. Em (a) temos uma rede não-dirigida e em (b) uma rede dirigida. No caso (a), os elementos a_{ij} da matriz são iguais a 1 se há uma ligação entre os vértices i e j e iguais a zero, caso contrário. Já no caso (b), os elementos da matriz a_{ij} são iguais a 1 se existe uma conexão dirigida do vértice i para o vértice j.

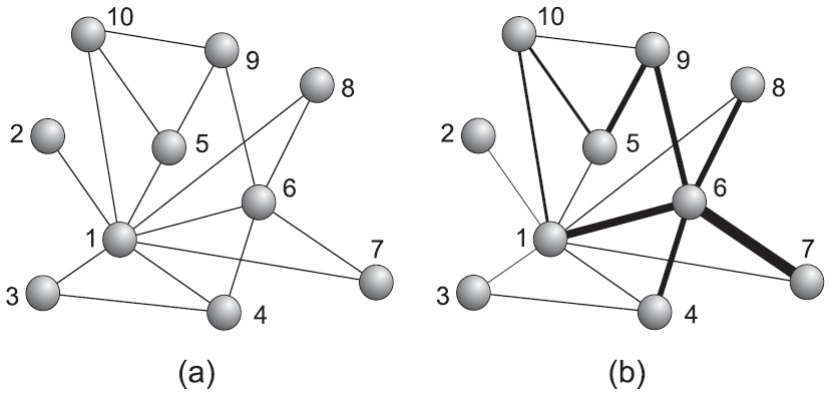

Figura 3.2: Exemplo de duas redes constituídas de 10 vértices e 15 arestas, sendo a rede em (a) não ponderada e a rede em (b) uma rede ponderada.

3.2 Métodos de reconhecimento de comunidades

O problema de reconhecimento em redes complexas é do tipo NP-difícil, sendo necessária a aplicação de alguma heurística para guiar a busca pelo espaço de soluções. A solução mais aceita atualmente pela comunidade cientifica, consiste na otimização da função de modularidade. Esta compara uma partição com uma rede aleatória. Sendo o grau do vértice i é dado por:

$$k_i = \sum_j A_{ij} \tag{3.1}$$

E a probabilidade que exista uma aresta entre os vértices i e j se as conexões forem realizadas de maneira aleatória, porém respeitando o grau dos vértices é dada por: $k_i k_j / 2m$, onde m é o número de arestas presente no grafo, $m = \frac{1}{2}\sum_{i,j} A_{ij}$. A medida de modularidade é definida

por:

$$Q = \frac{1}{2M}\sum_{i\neq j}(A_{ij} - \frac{k_i k_j}{2M})\delta(c_i, c_j) \tag{3.2}$$

onde $\delta(\sigma_i, \sigma_j)$ é igual a 1 se $c_i = c_j$ e igual a 0 caso contrário e c_i é a comunidade ao qual o vértice pertence. Se a fração de arestas entre comunidades não é diferente do que se espera de uma versão aleatória da rede, o valor de Q será próximo de 0 e caso este valor seja próximo de 1 a rede é bastante modular, ou seja, possui uma forte estrutura de comunidades.

A generalização desta medida para redes ponderadas é natural (NEWMAN, 2004a; FORTUNATO, 2010). Esta é dada pela expressão:

$$Q_w = \frac{1}{2W}\sum_{i\neq j}(W_{ij} - \frac{s_i s_j}{2W})\delta(c_i, c_j) \tag{3.3}$$

onde s_i é definido como a força dos vértices, dado por $s_i = \sum_j W_{ij}$ e W é a soma da força de todos as arestas da rede, ou seja, $W = \frac{1}{2}\sum_{i,j} W_{ij}$. Esta medida é análoga à apresentada anteriormente, equação 3.2 e a fração $\frac{s_i s_j}{2W}$ também constitui uma comparação com uma rede gerada aleatoriamente.

Nas seções à seguir serão apresentadas duas técnicas para otimização da modularidade. Ambas serão descritas utilizando a definição não ponderada da modularidade por simplicidade, porém, a generalização é natural. Além destas técnicas será apresentada uma técnica baseada em propagação de rótulos, que não tem como objetivo a maximização da modularidade.

3.2.1 Método Guloso

O algoritmo que utiliza uma heurística gulosa é apresentado em (NEWMAN, 2004b). Basicamente, inicia-se a partir de um conjunto de N vértices, sendo cada um considerado uma comunidade. A cada passo duas comunidades são combinadas de maneira que haja o maior crescimento da medida de modularidade Q, equação 3.2. Desta maneira, após $N-1$ iterações haverá apenas uma única comunidade e o algoritmo chega fim. Este processo gera um dendrograma, semelhante ao observado na seção 2.3. O dendrograma deverá ser segmentado no ponto onde há o maior valor da modularidade. A figura 3.3 é um exemplo de dendrograma obtido por este método, sendo que este já está segmentado segundo o valor de máxima modularidade.

Para simplificar a descrição do algoritmo, considere:

$$e_{ij} = \frac{1}{2m}\sum_{vw} A_{vw}\delta(c_v, i)\delta(c_w, j) \tag{3.4}$$

onde e_{ij} é a fração das arestas que unem vértices da comunidade i à vértices da comunidade j

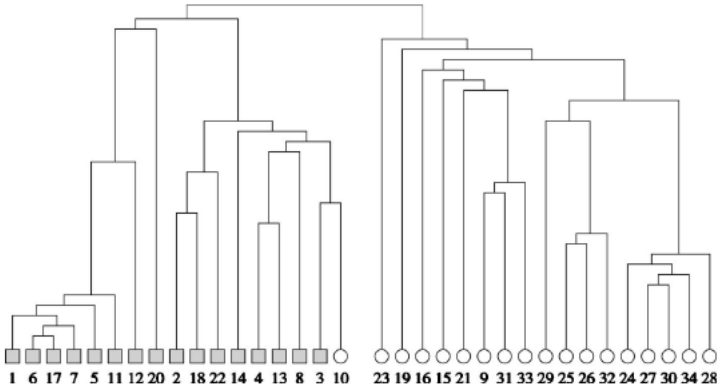

Figura 3.3: Dendrograma de comunidades encontrado pelo algoritmo guloso (*FastGreedy*) para a rede do clube de caratê de Zachary (GIRVAN; NEWMAN, 2002; ZACHARY,). As formas dos vértices representam os dois grupos presentes na quebra da rede devido a uma disputa interna no clube. Figura retirada de (NEWMAN, 2004b).

e:

$$a_i = \frac{1}{2m}\sum_v k_v \delta(c_v, i) \tag{3.5}$$

onde a_i é a fração das extremidades das arestas ligadas à vértices na comunidade i. Assim, escrevendo $\delta(c_v, c_w) = \sum_i \delta(c_v, i)\delta(c_w, j)$, obtém-se, da equação 3.2:

$$Q = \frac{1}{2M}\sum_{vw}\left[A_{vw} - \frac{k_v k_w}{2M}\right]\sum_i \delta(c_v, i)\delta(c_v, i) =$$

$$= \sum_i \left[\frac{1}{2M}A_{vw}\delta(c_v, i)\delta(c_v, i) - \frac{1}{2M}\sum_v k_v \delta(c_v, i)\frac{1}{2M}\sum_w k_w \delta(c_w, i)\right] =$$

$$= \sum_i (e_{ii} - a_i^2) \tag{3.6}$$

A combinação de pares de comunidades que não possuem arestas que as conecte nunca resultará em um aumento da modularidade, deve-se apenas considerar aqueles pares que possuem estas arestas, que será no máximo M, onde M é o número de arestas no grafo. A variação da modularidade pela junção de duas comunidades é dada por:

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j) \tag{3.7}$$

que é calculado em tempo constante.

Cada etapa do algoritmo é realizada em tempo constante $O(n)$. Assim cada etapa gasta, no pior caso $O(n+m)$. Há no máximo $n-1$ iterações para construção do dendrograma, assim o algoritmo é executado em $O((m+n)n)$, ou $O(n^2)$ em grafos esparsos.

A generalização desta abordagem para grafos ponderados é natural, considerando a força do nó ($s_i = \sum_j W_{ij}$) ao invés do grau e substituindo M por $W = \frac{1}{2}\sum_{i,j} W_{ij}$. Desta maneira obtêm-se a equação 3.3.

A metodologia apresentada aqui é a mesma apresentada no artigo (NEWMAN, 2004b), sendo que uma otimização desta é apresentada em (CLAUSET; NEWMAN; MOORE, 2004), onde o autor sugere modificações que tornam este algoritmo computacionalmente mais eficiente, possibilitando sua aplicação em redes com elevado número de vértices.

3.2.2 Método baseado em mecânica estatística

Modelo de Potts

Em mecânica estatística , o modelo de Potts é uma generalização do modelo de Ising, que descreve a interação entre os spins em um cristal, bem como a interação destes em materiais ferromagnéticos. No modelo de Ising apenas dois estados são admitidos para os spins, -1 e 1. Já no modelo de Potts, admite-se qualquer quantidade finita q de estados.

Este modelo é capaz de definir variáveis macroscópicas que caracterizam o comportamento geral da evolução temporal do conjunto, utilizando as informações microscópicas de interação entre os muitos elementos que compõem o sistema,

A variável macroscópica de maior importância do sistema é o Hamiltoniano, que mensura a energia de um estado particular de um grafo $G = (V, E)$ com N nós, utilizando a informação de relação entre estados de nós conectados por arestas. Ele é definido por:

$$\mathscr{H} = -J \sum_{(i,j)\in E(G)} \delta_{\sigma_i \sigma_j} \tag{3.8}$$

onde J é a força de interação entre as partículas (acoplamento), E é o conjunto de arestas presente no grafo e σ_i denota o spin individual de cada nó, sendo um valor discreto entre 1 e q.

A equação 3.8 tende a um estado final cujos spins estão todos alinhados, porém se houver uma interferência externa, teremos:

$$\mathscr{H} = -J \sum_{(i,j)\in E(G)} \delta_{\sigma_i \sigma_j} + f(\sigma) \tag{3.9}$$

Nota-se que há q^N possíveis estados para o sistema, cuja função de probabilidade é dada por,

$$p(\sigma_l = \alpha) = \frac{\exp[-\beta \mathscr{H}(\sigma_{i\neq l, \sigma_l = \alpha})]}{\sum_{s=1}^{q} \exp[-\beta \mathscr{H}(\sigma_{i\neq l, \sigma_l = s})]} \tag{3.10}$$

sendo $\beta = \frac{1}{kT}$, onde T representa a temperatura do sistema e $k \approx 1.38x10^{-}23J/K$ é a constante de Boltzmann.

Há uma forte dependência do modelo com a probabilidade, equação 3.10. Sabe-se que os sistemas complexos geralmente possuem grande número de vértices, com diferentes spins para cada nó do sistema, e a probabilidade de um estado, dada pela equação 3.8, é próxima de zero. Assim deseja-se saber quais as características do sistema no estado fundamental, ou seja, obter o mínimo global do Hamiltoniano, equação 3.9. Isto é possível por meio de simulações computacionais, utilizando, por exemplo, o método de otimização *simulated annealing*, que será descrito a seguir.

Simulated Annealing

O conceito fundamental do *simulated annealing* (KIRKPATRICK; GELATT, 1983; LAARHOVEN; LAARHOVEN; AARTS, 1987; VIDAL, 1993) remonta a uma técnica experimental para crescimento de cristais com o menor número de defeitos possível. Isto funciona através do lento resfriamento da amostra de modo que defeitos e outras impurezas da rede do cristal possam ser corrigidas de modo que uma estrutura cristalina pura é obtida a baixas temperaturas. Neste processo a temperatura, que representa a energia cinética, tem de decrescer muito lentamente, caso contrário, a configuração molecular torna-se estática e o cristal torna-se imperfeito.

No método *simulated annealing*, mapeia-se o problema de otimização de um sistema físico considerado, identificando-se a função de custo com o Hamiltoniano $\mathscr{H}$ do sistema e as variáveis $\mathbf{x}$ da função de custo como os graus de liberdade físicos do sistema. Também, correspondendo a mudanças elementares $\mathbf{y} \rightarrow \mathbf{z}$ de variáveis, escolhe-se uma dinâmica para o sistema físico, que para este caso será a mudança dos spins na rede. Então o sistema é simulado utilizando-se métodos de Monte Carlo, sendo que em uma dada temperatura T os pesos estatísticos serão dados neste caso pelos fatores de Boltzmann:

$$P(\mathbf{x}) = \frac{1}{Z} exp\left(-\Delta\mathscr{H}(\mathbf{x})/k_B T\right) \tag{3.11}$$

onde Z é função de partição canônica $Z = \sum_{\{\mathbf{x}\}} exp\left(-\mathscr{H}(\mathbf{x})/k_B T\right)$. Mais precisamente, cada conjunto de posições de todos os átomos do sistema é ponderado de acordo com o fator de probabilidade de Boltzmann $e^{-E/k_B T}$, no qual E é a energia da dada configuração do sistema.

No caso de problemas de otimização, a energia E corresponde à função objetivo a ser otimizada. A cada interação, uma nova configuração (próxima à anterior) é pseudo-aleatoriamente

gerada. A energia desta nova configuração é então calculada e a diferença de energia ΔE entre o estado atual e a nova configuração é determinada. Caso esta variação seja positiva $(\Delta E > 0)$, a probabilidade desta nova configuração ser aceita é dada pelo fator de Boltzmann, caso contrário $P(\Delta E) = 1$ e a transição será aceita pois a nova energia é menor que a atual. No caso de variação $\Delta E > 0$, um número pseudo-aleatório é gerado a partir de uma distribuição uniforme e a transição será aceita caso o número escolhido seja menor ou igual ao fator de probabilidade de Boltzmann. Em altas temperaturas, esta probabilidade estará próxima de 1, o que faz com que muitas transições de acréscimo sejam aceitas. A aceitação de tais transições faz com que o algoritmo seja capaz de escapar de mínimos locais. À medida que a temperatura diminui, o número de transições aceitas também diminui. Como consequência, o sistema deve escapar de mínimos locais e possivelmente atingirá o mínimo global ao atingir a temperatura mínima $(T \to 0)$.

Reconhecimento de comunidades

Uma função que mensura a qualidade de um particionamento da rede deve satisfazer as seguintes especificações: (i) recompensar arestas internas de comunidades; (ii) penalizar a ausência de arestas entre nós de um mesmo grupo; (iii) penalizar arestas existentes entre diferentes grupos e (iv) recompensar a ausência de arestas entre diferentes grupos. Seguindo os conceitos introduzidos pela equação 3.8 e 3.9, temos:

$$\mathscr{H}(\sigma) = -\sum_{i \neq j} a_{ij} A_{ij} \delta_{\sigma_i \sigma_j} + \sum_{i \neq j} b_{ij}(1 - A_{ij}) \delta_{\sigma_i \sigma_j} + \sum_{i \neq j} c_{ij} A_{ij}(1 - \delta_{\sigma_i \sigma_j}) - \sum_{i \neq j} d_{ij}(1 - A_{ij})(1 - \delta_{\sigma_i \sigma_j}) \tag{3.12}$$

onde A_{ij} denota o elemento (i, j) da matriz de adjacências, $\sigma_i \in 1, 2..., q$ denota o estado do spin i e as constantes a_{ij}, b_{ij}, c_{ij} e d_{ij} representa os pesos das contribuições individuais de cada termo, sendo o primeiro termo referente aos nós internos às comunidades, o segundo a ausência de nós internos, o terceiro os nós externos às comunidades e, por fim, o quarto, referente a ausência de nós externos. O número máximo de spins q determina o máximo número de comunidades permitido e, a principio, pode ser definido como N.

Caso as arestas e a ausência delas seja ponderada de maneira igualitária, ou seja, $a_{ij} = c_{ij}$ e $b_{ij} = d_{ij}$, resta apenas fazer uma escolha adequada para estes parâmetros, de preferência mantendo-se a dependência de um parâmetro. Como será mostrado a seguir uma escolha adequada é $a_{ij} = 1 - \gamma p_{ij}$ e $b_{ij} = \gamma p_{ij}$, onde p_{ij} é a probabilidade de haver uma aresta entre os nós i e j, normalizada de maneira que $\sum_{i \neq j} p_{ij} = 2M$. A partir desta escolha, a equação 3.12 pode

ser simplificada para:

$$\mathscr{H}(\sigma) = -\sum_{i \neq j}(A_{ij} - \gamma p_{ij})\delta_{\sigma_i \sigma_j} \tag{3.13}$$

A equação 3.13 é semelhante a equação 3.9, sendo o acoplamento entre os nós i e j dado por $J_{ij} = A_{ij} - \gamma p_{ij}$. O Hamiltoniano 3.13 deve ser comparado a distribuição de arestas do grafo em estudo, por meio de algum modelo. Com base neste conceito reescreve-se a equação 3.13:

$$\mathscr{H}(\sigma) = -\sum_{s}(m_{ss} - \gamma[m_{ss}]_{p_{ij}}) \tag{3.14}$$

onde m_{ss} representa o número de arestas internas da comunidade s e o símbolo $[\cdot]_{p_{ij}}$ é o valor esperado assumindo uma distribuição p_{ij}.

Ao comparar com uma distribuição aleatória, onde a probabilidade de conexão é igual para qualquer par de nós, $p_{ij} = p$, têm-se:

$$[m_{ss}]_p = p\frac{n_s(n_s-1)}{2} \tag{3.15}$$

onde n_s é o número de spins no estado s. Assim o Hamiltoniano 3.14 é reescrito como:

$$\mathscr{H}(\sigma) = -\sum_{i,j \in E}\delta_{\sigma_i \sigma_j} + \gamma p \sum_{s}^{q}\frac{n_s(n_s-1)}{2} \tag{3.16}$$

A equação 3.16 foi originalmente proposta em (REICHARDT; BORNHOLDT, 2004), sendo seu estudo melhor demonstrado em (REICHARDT; BORNHOLDT, 2006).

Caso o Hamiltoniano 3.13 seja comparado com um modelo nulo, ou seja, cuja distribuição de grau é preservada, mas as arestas são distribuídas aleatoriamente, têm-se:

$$p_{ij} = \frac{k_i k_j}{2M} \tag{3.17}$$

$$\mathscr{H}(\sigma) = -\sum_{i \neq j}(A_{ij} - \gamma\frac{k_i k_j}{2M})\delta_{\sigma_i \sigma_j} \tag{3.18}$$

Ao fazer $\gamma = 1$, é possível comparar o Hamiltoniano 3.18 com a equação 3.2 de modularidade de Newman e Girvan (NEWMAN; GIRVAN, 2004). Por meio desta comparação nota-se:

$$Q = -\frac{1}{M}\mathscr{H}(\sigma) \tag{3.19}$$

assim, ao se minimizar o Hamiltoniano 3.18 maximiza-se a modularidade, equação 3.2.

Tanto para o Hamiltoniano 3.16 quanto para o Hamiltoniano 3.18 o problema de reconhecimento da estrutura de comunidades é dado pela minimização desta grandeza por qualquer

método de otimização, como por exemplo o simmulated annealing, descrito anteriormente. Como apenas informações locais sobre os estados são necessárias, isto faz com que o processo seja de fácil implementação, além de ser paralelizável.

3.2.3 Método baseado em propagação de rótulos

A grande dificuldade dos métodos baseados na otimização da modularidade é seu custo computacional, o que torna esta abordagem inviável para redes de grande porte. Em (RAGHAVAN; ALBERT; KUMARA, 2007), os autores propõe um método alternativo, baseado na propagação de rótulos e que possui uma baixa complexidade de tempo.

Suponha que um nó x tenha k vizinhos, $x_1, x_2, ..., x_k$ e cada vizinho possui um rótulo indicando a qual comunidade ele pertence. Assim o rótulo de x será determinado pelo rótulo de seus vizinhos. Assume-se que cada vértice da rede pertence à mesma comunidade que a maioria de seus vizinhos. Cada nó é inicializado com um rótulo único, deixando que os rótulos se propaguem pela rede subsequentemente. Conforme os rótulos são propagados, grupos de nós densamente conectados entram em consenso com mesmo rótulo. Quando muitos destes grupos densos surgem eles continuam expandindo o máximo possível. Ao final deste processo, vértices que possuem mesmos rótulos são ditos como pertencentes à mesma comunidade.

Este processo é feito de maneira iterativa, sendo que a cada etapa, todo nó da rede tem seu rótulo atualizado de acordo com os rótulos de seus vizinhos. Isto pode ser feito de duas maneira, síncrona ou assíncrona. No primeiro caso, síncrono, o rótulo de um nó x em um tempo t é determinado a partir de uma função dos seus vizinhos no tempo $t-1$, ou seja, $C_x(t) = f(C_{x_1}(t-1), ..., C_{x_k}(t-1))$, onde $C_x(t)$ é o rótulo do nó x no tempo t. Já para o caso assíncrono a atualização é dada por uma função dos rótulos que já foram atualizados na presente iteração e dos que ainda não foram atualizados, ou seja, $C_x(t) = f(C_{x_{i1}}(t), ..., C_{x_{im}}(t), C_{x_{i(m+1)}}(t-1), C_{x_{ik}}(t-1))$, onde os vizinhos $x_{i1}, ..., x_{im}$ já tiveram seus nós atualizados na presente iteração e $x_{i(m+1),...,x_{ik}}$ que ainda não foram atualizados no tempo t. A segunda alternativa, assíncrona, reduz a probabilidade da oscilação de rótulos, que podem ocorrer em grafos com formato de estrela. A ordem em que os N vértices são atualizados é aleatória. Deve-se observar que ao inicio do processo tem-se N rótulos distintos e a cada iteração este número é reduzido, resultando, em apenas um rótulo para cada comunidade.

Idealmente este processo continua até que não haja mudança nos nós da rede, porém, é possível que alguns nós da rede possuam um número máximo de vizinhos em mais de uma comunidade. Assim, escolhe-se aleatoriamente o rótulo deste nó entre os possíveis. Desta maneira, este processo iterativo é realizado até que todo nó da rede possua o mesmo rótulo que a

maioria de seus vizinhos. Ao final, obtém-se um particionamento da rede em conjuntos disjuntos (comunidades). Seja $C_1, ..., C_p$ os rótulos ativos na rede e $d_i^{C_j}$ o número de vizinhos de i com rótulo C_j o algoritmo termina quando, pra todo nó i, se i tem rótulo C_m, então $d_i^{C_m} \leq d_i^{C_j}, \quad \forall j$. Todo nó que possui o mesmo rótulo é agrupado na mesma comunidade.

O algoritmo é descrito pelas etapas a seguir:
(i) Inicialize cada nó da rede com um rótulo. Para o nó x, $C_x(0) = x$;
(ii) Faça $t = 0$;
(iii) Organize aleatoriamente todos os nós da rede em um conjunto X;
(iv) Para cada $x \in X$, faça $C_x(t) = f(C_{x_{i1}}(t), ..., C_{x_{im}}(t), C_{x_{i(m+1)}}(t-1), C_{x_{ik}}(t-1))$. Onde f retorna o rótulo que ocorre com maior frequência entre os vizinhos.
(v) Se cada nó possui o mesmo rótulo que a maioria de seus vizinhos termina-se o algoritmo, caso contrario faça $t = t + 1$ e vá para etapa (iii);

Como cada nó é iniciado com um rótulo diferente, as primeiras iterações resultam em diversos grupos, pequenos e densos, de nós em consenso (com mesmo rótulo). Estes grupos agregam mais nós e começam a crescer, até atingirem a borda de outros grupos de consenso. Assim, eles começam a competir entre si por mais membros. Note que o algoritmo converge quando há um consenso entre os grupos. Deve-se ressaltar, também, que uma rede com apenas uma comunidade também é possível, como seria o caso de uma rede completamente aleatória, como no modelo de Erdös - Rényi (ERDOS; RENYI, 1959) que não possuem estruturas de comunidades.

O critério de parada do algoritmo não visa a maximização da modularidade, como os demais algoritmos apresentados anteriormente e, consequentemente, não há uma única solução para o problema.

Apesar de não buscar a otimização da modularidade, equação 3.2, este método apresentada resultados com altos valores para a mesma, como mostrado em (RAGHAVAN; ALBERT; KUMARA, 2007). Isto é um bom indicativo que o particionamento obtido por este método é significativo. O grande diferencial deste método em relação aos demais apresentados é possuir uma complexidade de tempo próxima à linear, permitindo a sua utilização em redes de grande porte, além de não ser necessária a informação do número de comunidades *a priori*.

3.3 Medidas de similaridade e dissimilaridade

Há muitas maneiras para se determinar quão similares ou dissimilares dois objetos de um conjunto de dados são. A seguir é feita a formulação matemática deste tipo de medidas e são apresentadas as algumas medidas encontradas na literatura (BISHOP, 2006; THEODORIDIS; KOUTROUMBAS, 2003). Dentro do contexto de redes complexas, quanto maior o peso de uma aresta maior a interação entre os vértices conectados pela mesma. Assim, apenas as medidas de similaridade serão possíveis para as técnicas propostas. Entretanto, é possível obter a similaridade à partir de uma medida de dissimilaridade aplicando-se a função exponencial negativa desta, como será mostrado nas seções à seguir.

3.3.1 Medidas de dissimilaridade

Medidas de dissimilaridade (DM, do inglês *dissimilarity measure*) são definidas como:

$$d : X \times X \to \mathbb{R}$$

onde, $\mathbb{R}$ é um conjunto de números reais,que

$$\exists \quad d_0 \in \mathbb{R} : -\infty < d_0 \leq d(\mathbf{x},\mathbf{y}) < +\infty, \quad \forall \mathbf{x},\mathbf{y} \in X \tag{3.20}$$

$$d(\mathbf{x},\mathbf{x}) = d_0, \quad \forall \mathbf{x} \in X \tag{3.21}$$

Se além disso,

$$d(\mathbf{x},\mathbf{x}) = d_0 \Leftrightarrow \mathbf{x} = \mathbf{y} \tag{3.22}$$

$$d(\mathbf{x},\mathbf{z}) \leq d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{z}), \quad \forall \mathbf{x},\mathbf{y},\mathbf{z} \in X \tag{3.23}$$

onde d é chamada *métrica de dissimilaridade*. A inequação 3.23 é conhecida como desigualdade triangular.

A mais conhecida medida de dissimilaridade é a distância euclidiana:

$$d_2(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{l}(x_i - y_i)^2} \tag{3.24}$$

assumindo valores no intervalo $[0,\infty)$, sendo esta é um caso especial da distância Minkowski, que é definida por:

$$D_M^p = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \tag{3.25}$$

para $p = 1$, tem-se a distância Manhattan, para $p = 2$ tem-se a distância euclidiana e para o

limite $p \to \infty$ tem-se a distância de Chebyshev:

$$\lim_{p\to\infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} = \max_{i=1}^{n} |x_i - y_i|. \tag{3.26}$$

3.3.2 Medidas de similaridade

Medidas de similaridade (SM, do inglês *similarity measure*) são definidas como:

$$s : X \times X \to \mathbb{R}$$

em que,

$$\exists \quad s_0 \in \mathbb{R} : -\infty < s(\mathbf{x},\mathbf{y}) \leq s_0 < +\infty, \quad \forall \mathbf{x},\mathbf{y} \in X \tag{3.27}$$

$$s(\mathbf{x},\mathbf{x}) = s_0, \quad \forall \mathbf{x} \in X \tag{3.28}$$

Se além disso,

$$s(\mathbf{x},\mathbf{x}) = s_0 \Leftrightarrow \mathbf{x} = \mathbf{y} \tag{3.29}$$

$$s(\mathbf{x},\mathbf{y})s(\mathbf{y},\mathbf{z}) \leq [s(\mathbf{x},\mathbf{y}) + s(\mathbf{y},\mathbf{z})]s(\mathbf{x},\mathbf{z}), \quad \forall \mathbf{x},\mathbf{y},\mathbf{z} \in X \tag{3.30}$$

s é chamada de *métrica de medida de similaridade*.

A seguir são definidas algumas medidas possíveis:

Medida de similaridade cosseno

$$s_{cossine}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\| \mathbf{x} \| \| \mathbf{y} \|} \tag{3.31}$$

onde $\| \mathbf{x} \| = \sqrt{(\sum_{i=1}^{l} x_i^2)}$ é o módulo do vetor $\mathbf{x}$.

Esta medida é assume valores entre -1 e 1, sendo invariante a rotações, mas não a uma transformação linear.

Coeficiente de correlação de Pearson

$$r_{pearson}(\mathbf{x},\mathbf{y}) = \frac{{x_d}^T y_d}{\| x_d \| \| y_d \|} \tag{3.32}$$

onde $x_d = [x_1 - \overline{x}, ..., x_l - \overline{x}]^T$ é chamado vetor diferença, sendo que $\overline{x} = \frac{1}{l}\sum_{i=1}^{l} x_i$. $r_{pearson}(\mathbf{x},\mathbf{y})$ assume valores dentro do intervalo -1 e 1, nota-se também que esta não depende diretamente de $\mathbf{x}$ e $\mathbf{y}$, mas sim dos vetores diferença x_d e y_d.

A medida de dissimilaridade associada a este coeficiente é:

$$D(\mathbf{x},\mathbf{y}) = \frac{1 - r_{Pearson}(\mathbf{x},\mathbf{y})}{2} \tag{3.33}$$

que assume valores no intervalo [0, 1].

Medida de Tanimoto

$$s_T(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\| \mathbf{x} \|^2 + \| \mathbf{y} \|^2 - \mathbf{x}^T\mathbf{y}} \tag{3.34}$$

Assume valores no conjunto $(-\infty, 1]$.

Medida proposta por Fu

$$s_c(\mathbf{x},\mathbf{y}) = 1 - \frac{d_2(\mathbf{x},\mathbf{y})}{\| \mathbf{x} \| + \| \mathbf{y} \|} \tag{3.35}$$

onde d_2 é a distância euclidiana. Assume valores entre 0 e 1, sendo seu máximo quando **x** = **y** e minimo para **x** = - **y**.

Inverso da distância euclidiana

$$s(\mathbf{x},\mathbf{y}) = \frac{1}{d_2(\mathbf{x},\mathbf{y})} \tag{3.36}$$

Assume valores dentro do conjunto $[0, \infty)$, portanto não é uma medida de similaridade.

Exponencial

$$s_{exp}(\mathbf{x},\mathbf{y}) = \alpha exp\left(-\alpha D_x(\mathbf{x},\mathbf{y})\right) \tag{3.37}$$

onde α é um número real qualquer e $D_x(\mathbf{x},\mathbf{y})$ é uma medida de dissimilaridade.

3.4 Metodologia de agrupamento baseada em redes

A metodologia proposta neste trabalho, consiste em, dado um conjunto de dados $X = \{\vec{x}_1, ..., \vec{x}_N\}$ com N observações, onde cada vetor $\vec{x}_i = (x_{i1}, x_{i2}, ..., x_{im})$ deste conjunto é formado por m atributos, interpretá-lo como um grafo totalmente conectado, onde o peso de cada aresta é dado por uma medida de similaridade, como as que foram apresentadas na seção 3.3, segundo a metodologia descrita na seção 3.1.

Construída a rede complexa a partir do conjunto de dados e das métricas de similaridade, pode-se aplicar algum dos métodos de reconhecimento de comunidades, discutidos na seção 3.2, para obter-se os *cluster*. Sendo assim, cada comunidade obtida será interpretada como um *cluster*. Nesta etapa deve-se considerar as particularidades de cada abordagem. Ao se considerar o método guloso tem-se a possibilidade de informar qual o número de *clusters* e, assim, realizar o corte manual do dendrograma fornecido por este método. Em geral isto tende a reduzir o erro cometido, pois é uma informação extra que está sendo inserida. Além disso, é possível utilizar

a informação contida no dendrograma, de maneira semelhante a que ocorre nos algoritmos hierárquicos, como foi discutido na seção 2.3.

É importante perceber que a grande diferença entre a metodologia proposta utilizando o método de reconhecimento de comunidades guloso e o algoritmo de agrupamento hierárquico é que na abordagem proposta, a construção do dendrograma é guiada de maneira indireta pela similaridade, sendo guiada principalmente pela maximização da modularidade. Já o agrupamento hierárquico utiliza-se apenas a medida de similaridade e métrica deligação (*linkage metrics*).

Ao se utilizar o método baseado em mecânica estatística, não há a criação de um dendrograma e a informação relativa ao número de grupos só pode ser utilizada limitando-se o número de spins. Assim, o número de grupos existentes no particionamento será menor ou igual ao número de spins, que é definido previamente. É importante observar que quando não se conhece o número de grupos existentes na base de dados analisada deve-se informar um número de spins grande o suficiente.

A abordagem por meio da propagação de rótulos é a mais simples dentre as que foram discutidas neste capitulo, porém é a mais adequada se o conjunto de dados for grande, pois esta apresenta um custo computacional muito baixo, próximo ao linear em relação ao número de elementos do conjunto de dados.

A abordagem proposta pode ser utilizada como uma combinação de qualquer medida de similaridade com qualquer método de reconhecimento de comunidades, abrangendo, assim uma grande quantidade de possibilidades. É importante ressaltar que os resultados estarão sempre condicionados à escolha destes. No entanto, este problema está intimamente relacionado à maneira como os dados de entrada estão dispostos no espaço. Esta metodologia não resolve estes problemas, porém, apresenta bons resultados à diversas situações, se destacando dos métodos tradicionais em alguns casos, tais como as bases de dados íris e *wine*, entre outras que serão discutidas no próximo capitulo.

No próximo capitulo a metodologia de redes será comparada as demais descritas até aqui. Algumas questões relativas à robustez em relação à dimensionalidade dos dados, separação entre classes e densidade dos dados de entrada também serão analisadas. Estas analises serão feitas considerando-se bases de dados reais e sintéticas.

4 Resultados

4.1 Resultados e discussões

Neste capitulo será apresentada a comparação entre os métodos tradicionais de classificação não-supervisionada e a metodologia proposta. Serão avaliadas várias combinações de medidas de similaridade e métodos para reconhecimento de comunidades.

No inicio do capitulo 2 foram discutidas as diversas definições sobre agrupamento de dados presente na literatura. Assim, para avaliação do método proposto foram utilizadas duas bases de dados geradas artificialmente, sendo que cada uma se encaixa em uma categoria diferente. A primeira é constituída por duas gaussianas e é ideal para métodos baseados em centróide. Já a segunda é formada por duas meias luas e é ideal para definições que privilegiem a continuidade e a proximidade entre elementos de mesmo grupo.

Além disto, foram avaliadas as bases de dados Íris e Wine, disponíveis no repositório da UCI (ASUNCION; NEWMAN, 2007), sendo a primeira uma das mais populares bases de dados no contexto de reconhecimento de padrões.

4.1.1 Bases sintéticas

Para validação da metodologia de classificação não-supervisionada baseada em redes complexas, utilizou-se dois conjuntos de bases de dados gerados artificialmente. O primeiro é composto por dois conjuntos de pontos distribuídos em um espaço bidimensional segundo duas gaussianas. Cada gaussiana é composta por 500 pontos. Sendo assim, o conjunto de dados é formado por 1000 pontos. Além disto, cada distribuição possui matriz de covariância dada pela matrix identidade e a distância entre suas médias varia de $d = 0$ a $d = 15$. As figuras 4.1 (a) a (c) exemplificam este procedimento para três distâncias, $d = 0, 3$ e 15. Observe que conforme d cresce, a identificação dos grupos torna-se mais simples. O segundo conjunto de dados é um dos problemas clássicos de reconhecimento de padrões (THEODORIDIS; KOUTROUMBAS, 2003), consisitindo de dois conjuntos de pontos uniformemente distribuídos em duas áreas limitadas por semi-círculos. Neste caso varia-se o número de pontos distribuídos nestas áreas, ou seja, a densidade de pontos nas regiões. Variou-se tal densidade de $\rho = 1$ até $\rho = 32$ em passos de 1.6, ou seja, de 200 a 5960 ponto, em passos de 320 pontos. A figura 4.1 (d) a (f) mostra a configuração desta base de dados para três diferentes densidades, $\rho = 1, 6.4$ e 14.4.

A consistência do agrupamento obtido nestas bases de dados será avaliada de acordo com o índice de Jaccard, descrito na seção 2.7.

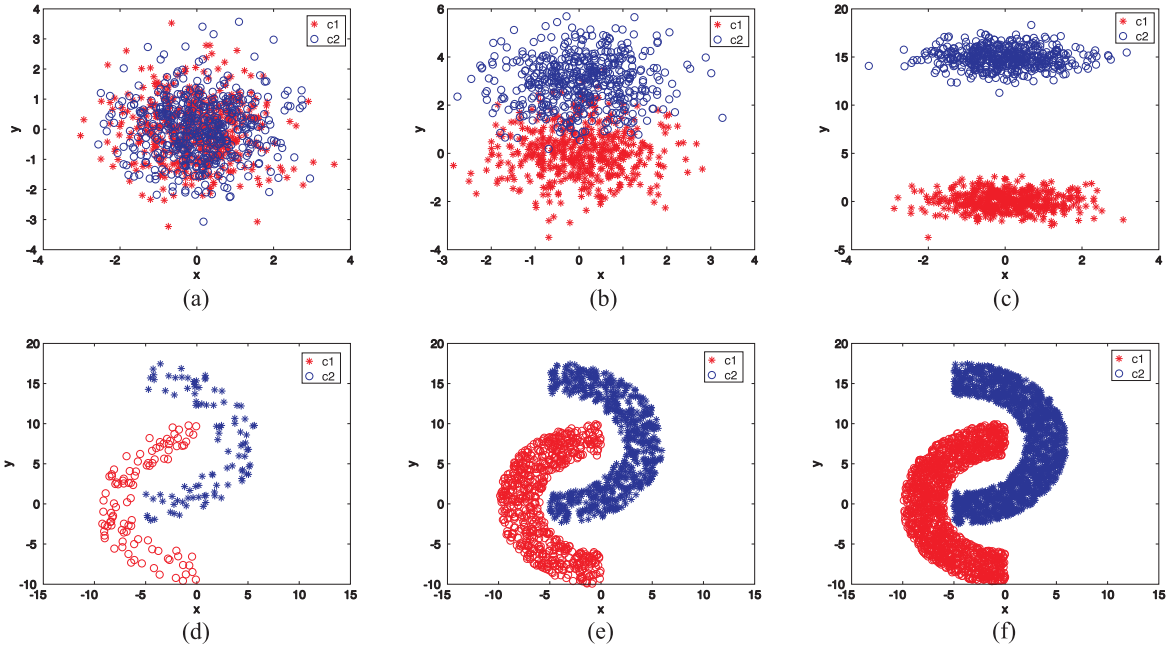

Figura 4.1: Exemplos de bases de dados sintéticas utilizadas para avaliar o método de agrupamento proposto. Os pontos do primeiro conjunto foram gerados de acordo com distribuições gaussianas, na qual as médias são separadas pelas distâncias (a) $d = 0$, (b) $d = 3$ e (c) $d = 15$. O segundo conjunto é formado por duas meias luas e varia-se a densidade dos pontos, obtendo-se *clusters* mais bem definidos. Nos exemplos utilizou-se as densidades (a) $\rho = 1.0$, (b) $\rho = 6.4$ e (c) $\rho = 14.4$. Figura adaptada de (RODRIGUES; ARRUDA; COSTA, 2011).

4.1.2 Resultados obtidos

Bases gaussianas

O problema das duas bases de dados é bastante simples para algoritmos como k-médias 2.2 e *Expectation Maximization* (EM) 2.6. No primeiro método, os centróides convergirão para valores próximos as médias das gaussianas e os trechos onde há a sobreposição das duas distribuições será dividido igualmente, ou seja, o erro será igualmente distribuído entre os grupos, fazendo que os índices de validação melhorem. Conforme a distância entre as médias creste o problema fica cada vez mais simples, e o erro diminui cada vez mais. Como trata-se de duas distribuições gaussianas o algoritmo EM apresenta um resultado semelhante ao resultado obtido pelo método k-médias (Esta semelhança é apresentada na seção 2.6, ver figura 2.3). Assim, a comparação com a metodologia proposta limitou-se apenas ao k-médias e ao algoritmo CobWeb, que apresenta um viés diferente do primeiro algoritmo. As simulações, desta seção, foram realizadas dez vezes, pois o procedimento de criação das bases de dados envolve fatores aleatórios. Desta maneira todos os gráficos são um média destas simulações.

É importante observar que quando as distâncias entre as médias das duas gaussianas é

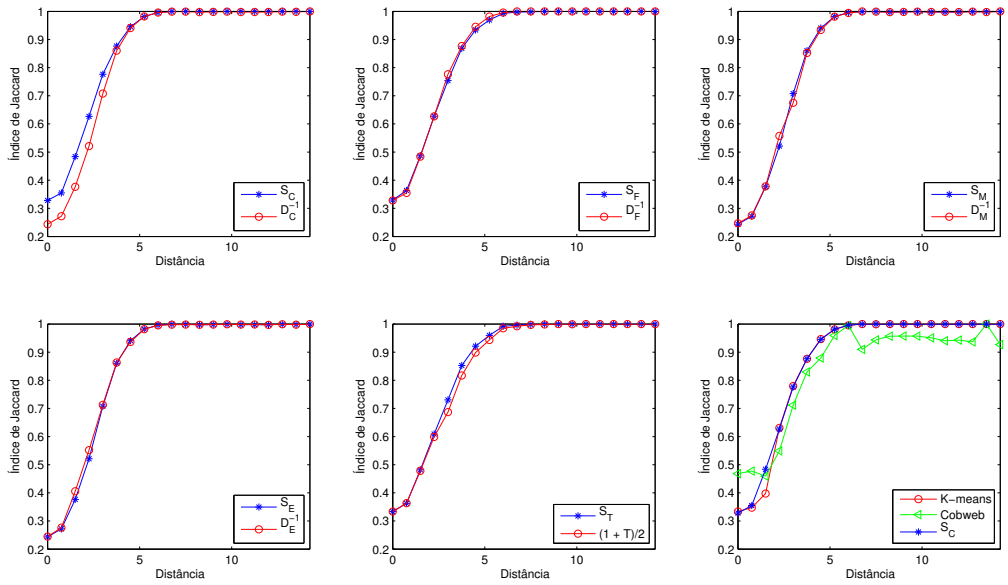

Figura 4.2: Índices de Jaccard obtidos em função da separação dos *clusters* para o conjunto de pontos gerados por duas distribuições gaussianas em um espaço bi-dimensional. Veja figura 4.1 (a) - (c). Os melhores resultados para método de reconhecimento guloso (*FastGreedy*) e cada medida de similaridade: (a) Distância de Chebyshev, (b) Similaridade de Fu, (c) Distância Manhattan, (d) Distância Euclidiana e (e) Similaridade de Tanimoto. O número de *clusters* foi determinado automaticamente pelo máximo valor da modularidade em todos os casos. No item (f) o método baseado em redes é comparado com melhores abordagens de agrupamento, neste caso, k-médias e cobweb. Cada ponto é uma média de 10 execuções. Figura adaptada de (RODRIGUES; ARRUDA; COSTA, 2011)

muito próximas de zero não há grupos formados (ver figura 4.1) e algoritmos como k-médias identificam estes grupos, pois eles possuem a informação, a priori, do número de *clusters*.

O método baseado em redes foi avaliado nas bases gaussianas considerando-se diversas medidas de similaridade. No caso da utilização do método de detecção de comunidades guloso, não se utilizou qualquer limiar. Optou-se pela utilização deste método, pois a base de dados é relativamente pequena e este algoritmo tende a apresentar bons resultados (NEWMAN, 2004b). Além disso, dentre os métodos de reconhecimento de comunidades este é o único que permite determinar, efetivamente, o número de grupos. Apesar disto, esta informação não foi utilizada nos gráficos da figura 4.2, pois o erro apresentado pelo corte no máximo valor de modularidade apresentou resultados muito semelhantes aos obtidos pelo corte, no dendrograma, para obtenção de dois grupos.

A figura 4.2 apresenta os resultados obtidos para as melhores medidas de similaridade. Cada gráfico apresenta uma medida de similaridade e sua reepectiva exponencial. Sendo elas:

(a) Distância de Chebyshev, (b) Similaridade de Fu, (c) Distância Manhattan, (d) Distância Euclidiana e (e) Similaridade de Tanimoto. No gráfico do item (e) é feita a comparação do método que apresentou os maiores valores do índice de Jaccard da metodologia baseada em redes com os métodos tradicionais k-médias e cobweb. A análise desta figura mostra que os tanto o algoritmo proposto, quanto o k-médias apresentam resultados semelhantes. Nos primeiros pontos deste gráfico, nota-se que o algoritmo CobWeb apresenta um desempenho superior aos outros dois, isto ocorre, pois ele obtêm como resultado um *cluster* muito grande, o que na realidade não deve ser considerado um erro, porém o índice de Jaccard leva isto em consideração. O desempenho inferior deste método, em comparação com os outros dois, em pontos onde a separação é razoável é explicada pois este método encontra mais que dois grupos, degradando seu desempenho.

Para este tipo de problemas o método baseado em redes apresentou resultados bastante satisfatórios, já que seu índice de Jaccard é comparável aos obtidos com o k-médias, que é o método mais adequado para este tipo de base de dados. Porém é importante ressaltar que na metodologia proposta não foi necessário fornecer a informação relativa ao número de grupos presentes, enquanto que o k-médias se utilizou desta informação.

No particionamento de problemas reais, geralmente deseja-se reduzir ao máximo o custo computacional do processo de agrupamento. A possibilidade exposta no capitulo 3 foi a aplicação de um valor de limiar para cada nó, mantendo a rede conexa, como discutido anteriormente. Aqui deseja-se avaliar qual o impacto desta metodologia nos resultados do agrupamento. Assim executou-se o método de agrupamento baseado em redes para a medida de similaridade exponencial da distância de Chebyshev em função da distância entre as médias de maneira semelhante à realizada no experimento anterior variando-se os valores de limiar. Aqui, novamente, utilizou-se o método guloso de reconhecimento de comunidades, devido as vantagens apresentadas acima. Neste caso, porém, utilizou-se os resultados de erro, índice de Jaccard e modularidade para grupos determinados automaticamente, figura 4.3 nos itens (a) a (c), e definiu-se o número de grupos nos itens (d) a (f). Escolheu-se a medida de similaridade exponencial da distância de Chebyshev, pois esta apresentou os melhores resultados no experimento anterior.

A análise dos gráficos da figura 4.3 mostram que é possível aplicar o limiar proposto e obter resultados muito próximos dos obtidos sem a aplicação de qualquer limiarização. Assim, concluí-se que apenas uma pequena fração das arestas implica de fato em um melhor resultado de agrupamento. Porém deve-se observar que o valor da modularidade tende a crescer para uma separação ruim, isto é, pequena distância entre as médias das gaussiana, em função do

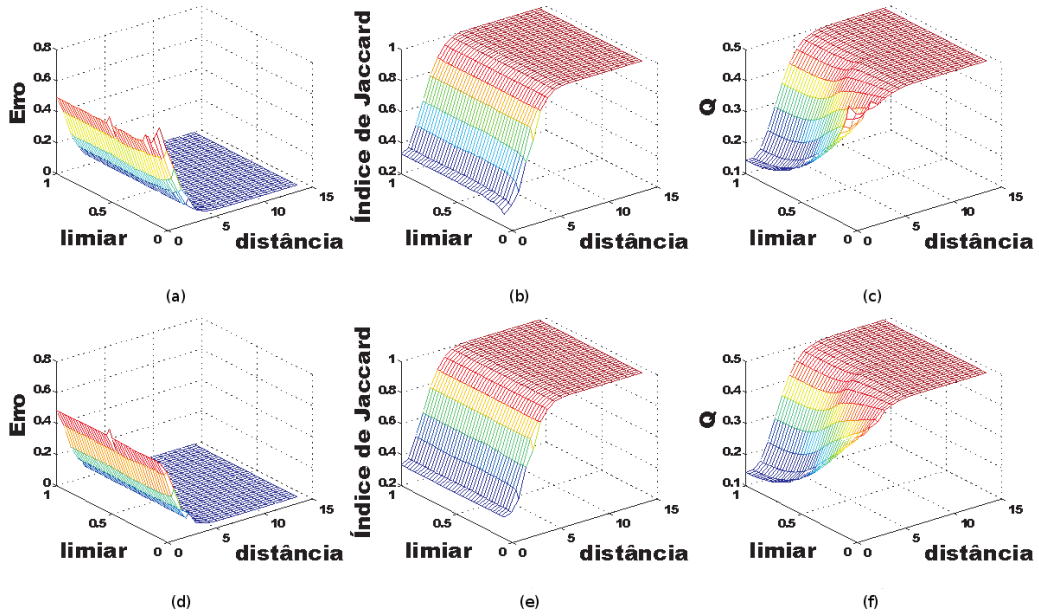

Figura 4.3: Índices de Jaccard, erros e valores de modularidade obtidos em função da separação dos *clusters* e do valor de limar para o conjunto de pontos gerados por duas distribuições gaussianas em um espaço bi-dimensional. Resultados obtidos para o método de reconhecimento guloso (*FastGreedy*) para a medida de similaridade exponencial da distância de Chebyshev. Em (a),(b) e (c) o número de grupos foi obtido automaticamente pelo máximo valor da modularidade. Em (d), (e) e (f) o número de grupos foi informado ao método, que realizou o corte adequado no dendrograma, ou seja, $k = 2$. Cada ponto é uma média de 10 execuções.

valor de limiar. Desta maneira deve-se evitar a utilização de valores de limiar muito grandes, mantendo-se uma boa parte da fração de arestas.

Outra análise importante é verificar o desempenho do método em relação a dados de alta dimensionalidade e como definir uma medida de similaridade adequada para estes problemas. A distância de Minkwoski, apresentada na seção 3.3, equação 3.25, é definida como uma função do parâmetro p. Assim, deseja-se saber qual o comportamento do método em relação à este parâmetro, bem como em função da dimensionalidade dos dados. Neste caso utilizou-se bases com dois conjuntos de pontos normalmente distribuídos, variando-se o número de dimensões do espaço e mantendo a distância entre as médias constante com $d = 4$. Os resultados são mostrados na figura 4.4.

Pela observação da figura 4.4 fica claro que para altas dimensões o desempenho do método tende a degradar e que escolhas de valores pequenos para p conduzem à resultados imprecisos. Assim uma boa escolha inicial é a exponencial da distância de Chebyshev, já que a queda de desempenho desta se dá apenas para valores de altas dimensionalidade. Aqui é importante ressaltar que o método proposto neste trabalho não é geral, resultando em bons resultados para

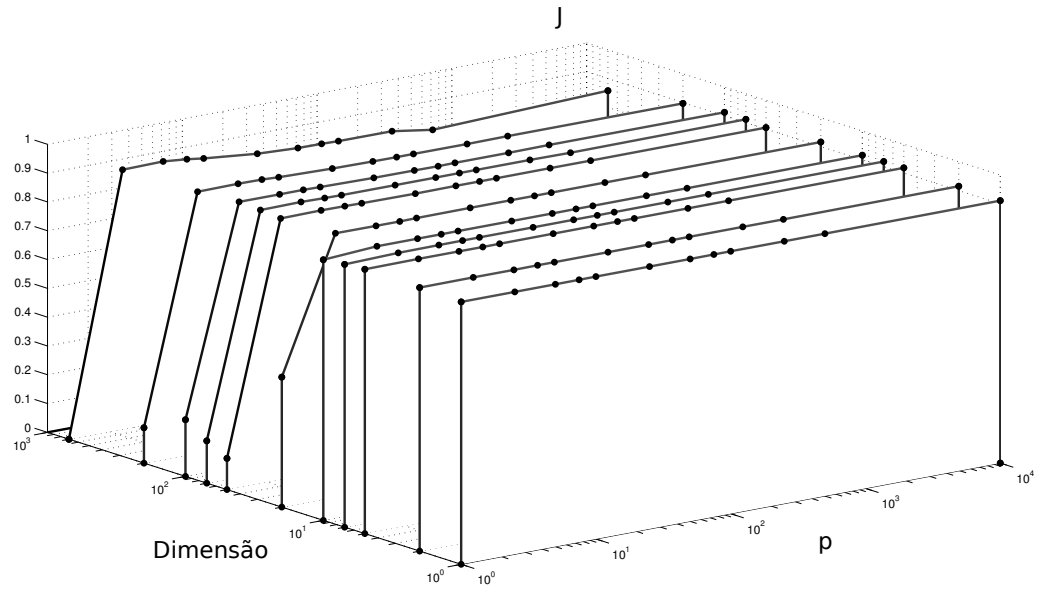

Figura 4.4: Índices de Jaccard obtidos em função da dimensionalidade do dado e do parâmetro p da exponencial da distância de Minkowski para o conjunto de pontos gerados por duas distribuições gaussianas com as médias separadas em uma distância $d = 4$. Os resultados foram obtidos utilizando-se o método de reconhecimento de comunidades *FastGreedy*. Veja figura 4.1. Cada ponto é uma média de 10 execuções.

todos os tipos de dados. Isso se deve ao fato de que o problema de reconhecimento de padrões é muito complexo, não havendo até mesmo um consenso da comunidade científica sobre o que realmente é um padrão. Logo, os métodos existentes apresentam bons resultados em alguns tipos de dados, não sendo gerais. O mesmo ocorre com o método presente. No entanto, vale observar que os métodos que apresentam resultados precisos em dados com estrutura modular são métodos adequados para serem usados na identificação de padrões.

Bases de duas meias luas

O problema das duas meias luas é clássico em reconhecimento de padrões. Métodos como o k-médias, que na seção anterior apresentavam resultados precisos em dados com distribuição gaussiana, não tem desempenho satisfatório nesta classe de problemas. Como dito no capitulo 2 o k-médias tende a formar grupos com formatos esféricos, assim, seu desempenho é muito baixo para esta classe de problemas. A figura 4.1.2 mostra a comparação do método baseado em redes com o k-médias. Fica claro que o método proposto é bem superior ao k-médias para esta situação. Entretanto é necessário considerar a densidade dos pontos neste tipo de situação, principalmente quando se analisa métodos hierárquicos ou baseados nestes. Pe-

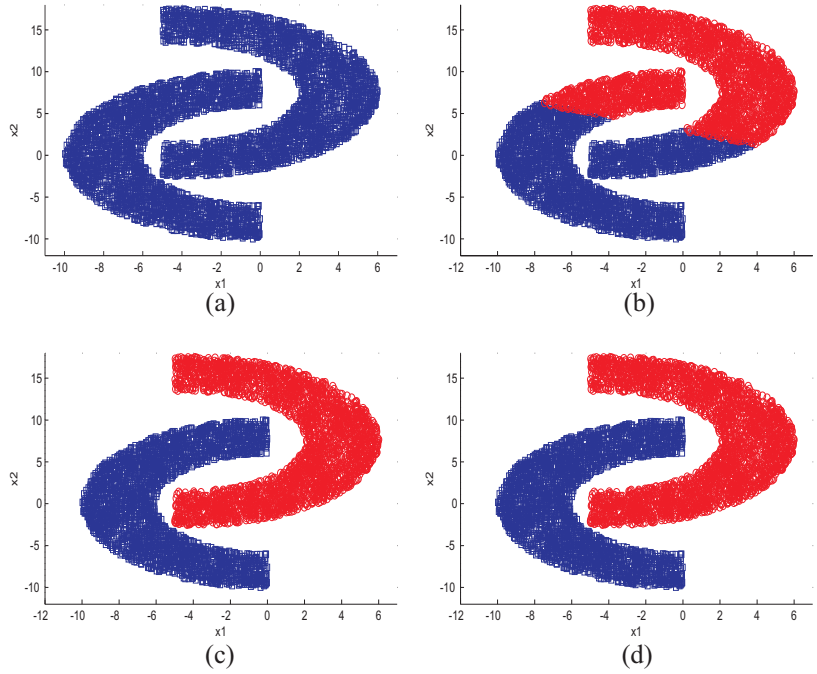

Figura 4.5: Exemplo do melhor resultado obtido para (b) k-médias e método baseado em redes usando máximo valor da modularidade em (c) e fixando $k = 2$ e (d). O conjunto de dados original é mostrado em (a). Figura adaptada de (RODRIGUES; ARRUDA; COSTA, 2011).

las análises realizadas à seguir ficará claro que o método proposto é bastante sensível a este parâmetro.

O método baseado em redes é bastante influenciado por seus vizinhos mais próximos, como ficou claro pela análise feita a partir da limiarização no item anterior. Sendo assim, é importante analisar o desempenho deste variando-se a densidade de pontos. Para realizar esta operação, optou-se pela base de duas meias luas. Verifica-se, pela observação da figura 4.1, que em alguns casos pode haver a formação de pequenos "*sub-clusters*", principalmente quando a densidade é pequena. Assim, espera-se que o desempenho do método aumente conforme aumenta-se a densidade de pontos. Isto é observado na figura 4.6.

A figura 4.6 mostra o desempenho do método ao se variar a desnsidade para o método de reconhecimento de comunidades guloso e as melhores medidas de similaridade, que neste caso foram a exponencial da distância de Chebyshev, exponencial da distância euclidiana, exponencial da distância Manhattan, inverso da distância de Chebyshev, inverso da distância euclidiana e inverso da distância Manhattan. Nesse caso, não foi mostrado o gráfico comparando com os métodos tradicionais, pois tanto o k-médias quanto o CobWeb apresentam resultados aproximadamente constantes, ou seja, não apresentam ganho de desempenho ao se aumentar a densidade. Para o algoritmo k-médias isto é bastante natural, devido ao viés de seu modelo, sendo seu erro

médio próximo de 15%. O problema presente na abordagem do método CobWeb é que este cria muitos *clusters*, fazendo com que o erro do método cresça muito, sendo este erro com valor médio próximo de 85%.

O método baseado em redes apresentou um resultado bastante satisfatório para grande parte das medidas de similaridade. Sendo que os melhores resultados foram obtidos pelas exponenciais das distâncias de Chebyshev, figura 4.6 (a), euclidiana, figura 4.6 (b) e Manhattan, figura 4.6 (c). Nota-se, também, que não há uma divergência muito grande entre os resultados obtidos por meio do corte no dendrograma correspondendo a dois grupos e o corte de máxima modularidade. Isto é mais um indicativo que a maximização da modularidade é uma função adequada para guiar a escolha relativa ao número de *clusters*, já que os resultados para bases gaussianas, da seção anterior, também é positivo e está em concordância com estes.

Assim, dentre as possibilidades analisadas aqui, o método proposto mostrou-se superior para o problema das duas meias luas, visto que obteve acerto de 100% em determinados casos, como verificado na figura. Também fica claro pela observação dos gráficos da figura 4.6, que, para algumas medidas de similaridade, a barra de erro toca o índice de Jaccard igual à 1, indicando um agrupamento perfeito.

4.1.3 Bases de dados reais

Até aqui foram feita apenas análises levando em consideração bases de dados sintéticas, onde a distribuição dos dados era gaussiana ou uniforme. Nesta seção deseja-se avaliar o desempenho para bases de dados reais, onde haverá problemas com mais de dois grupos e classes desbalanceadas (na base de dados *Wine*). Além disto, as distribuições podem não ser triviais.

Serão analisadas duas bases de dados reais, Iris e *Wine*, sendo a ambas compostas por três classes e a primeira é balanceada, ou seja, todas as classes tem o mesmo número de elementos, enquanto a segunda é desbalanceada. Quanto a dimensionalidade, é importante enfatizar que a base de dados *Wine* possui uma dimensionalidade maior, constituindo mais uma dificuldade, devido a problemas relacionados à maldição da dimensionalidade (THEODORIDIS; KOUTROUMBAS, 2003; BISHOP, 2006).

Para as bases sintéticas, utilizou-se apenas o método de reconhecimento de comunidades guloso, nesta seção, porém, foram testados as demais técnicas apresentadas no capitulo 3, como método baseado em mecânica estatística, também chamado de *SpinGlass* e o método de propagação de rótulos, chamado aqui por *LabelPropagation*. Quanto as medidas de similaridade, enfatizou-se a distância de Minkowski, já que nesta é possível variar o parâmetro p

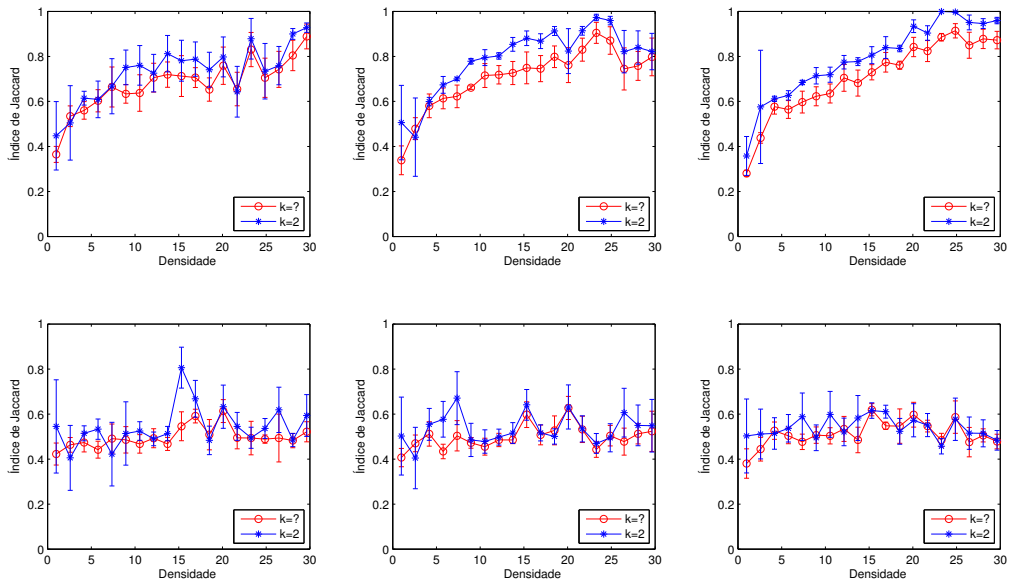

Figura 4.6: Índices de Jaccard obtidos em função da densidade dos *clusters* para o conjunto de pontos gerados por duas meias luas em um espaço bi-dimensional. Veja figura 4.1 (d) - (f). Foi utilizado o método de reconhecimento de comunidades guloso (*FastGreedy*). As medidas de similaridade utilizadas foram: (a) Exponencial da distância de Chebyshev, (b) Exponencial da distância Euclidiana, (c) Exponencial da distância Manhattan, (d) Inverso da distância de Chebyshev, (e) Inverso da distância Euclidiana e (f) Inverso da distância Manhattan. O número de *clusters* foi obtido automaticamente pelo máximo valor da modularidade e também informado manualmente $k = 2$ para comparação. Cada ponto é uma média de 10 execuções. Figura adaptada de (RODRIGUES; ARRUDA; COSTA, 2011).

e se ajustar ao espaço de características de maneira mais adequada, de maneira semelhante à realizada durante a análise da dimensionalidade no experimento com as bases gaussianas (Ver figura 4.4).

Íris

A base Íris (FISHER, 1936) é uma das bases de dados mais conhecidas na literatura de reconhecimento de padrões e aprendizado de máquina. É um conjunto de dados que contêm três classes, *Iris setosa*, *Iris virginica* e *Iris versicolor* com 50 elementos cada. Uma destas classes é linearmente separável das demais, enquanto as outras não (ASUNCION; NEWMAN, 2007). Esta base é descrita por quatro atributos comprimento e largura da pétala e sépala em centímetros. Para uma visualização desta, estes atributos foram projetados em duas dimensões, segundo a técnica de análise de componentes principais, PCA (THEODORIDIS; KOUTROUMBAS, 2003; BISHOP, 2006), sendo seu resultado mostrado na figura 4.7. Esta

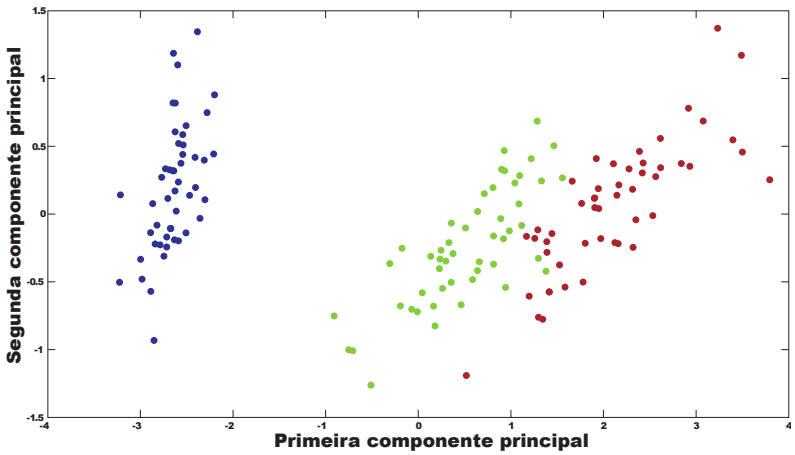

Figura 4.7: Projeção da base de dados Íris segundo a técnica de componentes principais, PCA.

técnica projeta os dados segundo os eixos onde há a máxima variância dos dados, não garantindo nada em relação a separação entre as classes.

A figura 4.8 mostra o procedimento de construção do dendrograma utilizado pelo método guloso utilizando-se a medida de similaridade exponencial da distância de Minkowski com parâmetro $p = 0.5$. Acima do dendrograma é mostrada a função de modularidade. A linha tracejada indica o corte para o valor onde a modularidade é máxima, com erro de 6.0% e, também pode-se observar o caso correspondente a dois grupos com um erro de 3,33%. Observa-se que o dendrograma fornecido pelo método contém informações importantes sobre a "configuração" dos elementos no espaço de características. Nota-se, que de fato, há uma classe que se separa melhor das outras duas, isto é verificado também pela projeção segundo a técnica PCA, na figura 4.7.

Foi feita a comparação entre a metodologia proposta e os métodos tradicionais, expostos no capitulo 2. Para a abordagem baseada em redes foram testadas várias combinações entre métodos de detecção de comunidades e medidas de similaridade. Os melhores resultados são mostrados na tabela 4.1. Nesta são mostrados os valores do erro obtido, bem como o índice de Jaccard. Utilizou-se também a informação relativa ao número de grupos, tanto para a metodologia baseada em redes quanto para as demais metodologias que apresentam esta flexibilidade.

Como os resultados para esta base de dados são piores, para todos os métodos analisados, quando se normaliza os dados, optou-se por utilizá-los sem normalização.

Pela análise da tabela 4.1 fica claro que a abordagem baseada em redes é bastante promissora, porém é importante observar a forte dependência do método em relação a métrica utilizada

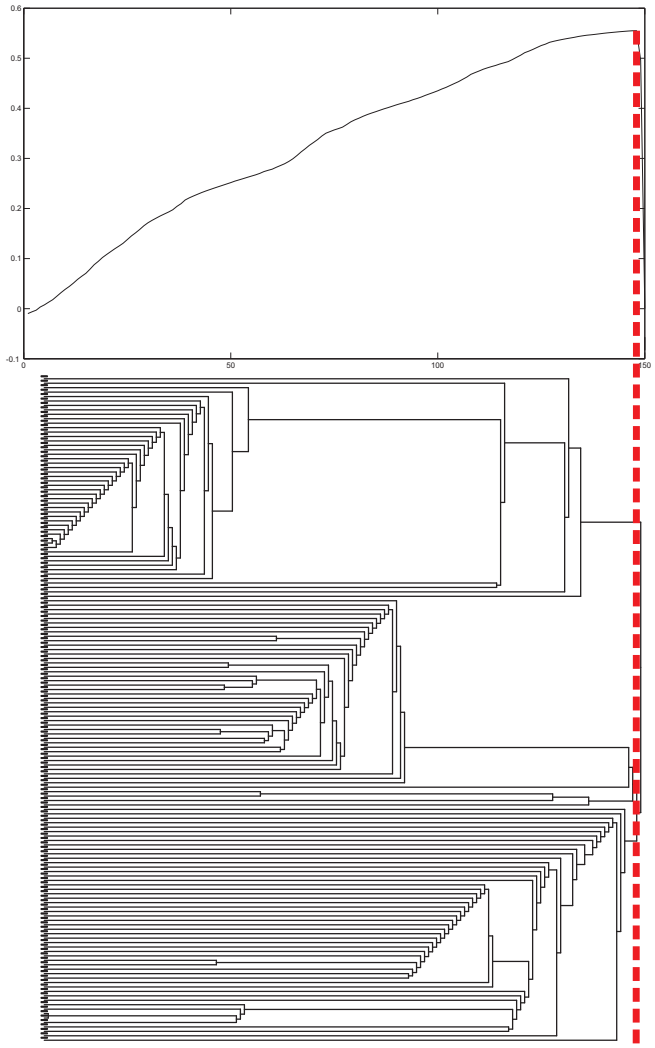

Figura 4.8: Dendrograma obtido pelo método guloso (*FastGreedy*), considerando a medida de similaridade exponencial da distância de Minkowski com parâmetro $p = 0.5$, para a base de dados Iris. O corte no dendrograma resulta em três classes, com um erro de 3,33%. Figura adaptada de (RODRIGUES; ARRUDA; COSTA, 2011).

no espaço de atributos. Observa-se que uma parte das configurações testadas apresenta erro de 33,3%, isto ocorre, pois o método identifica as classes não linearmente separáveis como um único grupo. Neste caso é importante observar os valores obtidos pelo índice de Jaccard e não apenas o erro, pois este fornecerá uma analise mais adequada a este problema. Pela análise deste índice, pode-se notar um fato interessante: para o método "$\exp(-M_{0.5})$ - *LabelPropagation*" obteve-se um erro de 15.33% e $J = 0.70$, já para o método "$\exp(-M_2)$ - *SpinGlass*" o erro foi menor, de 14.8%, porém o índice de Jaccard também foi menor $J = 0.63$, este exemplo mostra que a análise mais adequada deve ser feita pelo índice.

Ao comparar estes resultados com os métodos tradicionais, percebe-se que, mesmo nos piores casos da metodologia proposta, algoritmos bastante sofisticados, como o *Expectation*

Tabela 4.1: Erros obtidos para o agrupamento na base de dados Iris, considerando os casos onde o número de *clusters* é conhecido, $k = 3$, ou não, $k =?$. Os algoritmos EM, k-médias e *Farthest First* utilizam esta informação. A abordagem baseada em redes foi testada para as medida de similaridade baseadas na exponencial da distância de Minkowski e três métodos diferentes para reconhecimento da estrutura de comunidades, *FastGreedy*, *LabelPropagation* e *SpinGlass*

Method	% error ($k =?$)	% error ($k = 3$)	Jaccard ($k =?$)	Jaccard ($k = 3$)
k-means	–	11.3	–	0.68
cobweb	33.3	–	0.59	–
farthest first	–	14.0	–	0.63
EM	40.0	9.33	0.46	0.72
$\exp(-M_{0.5})$ - *FastGreedy*	6.0	3.33	0.83	0.87
$\exp(-M_1)$ - *FastGreedy*	33.33	30.66	0.59	0.58
$\exp(-M_2)$ - *FastGreedy*	33.33	32.66	0.595	0.59
$\exp(-M_\infty)$ - *FastGreedy*	33.33	32.0	0.58	0.58
$\exp(-M_{0.5})$ - *LabelPropagation*	15.33	–	0.70	–
$\exp(-M_1)$ - *LabelPropagation*	33.33	–	0.59	–
$\exp(-M_2)$ - *LabelPropagation*	33.33	–	0.59	–
$\exp(-M_\infty)$ - *LabelPropagation*	33.33	–	0.59	–
$\exp(-M_{0.5})$ - *SpinGlass*	28.93	–	0.58	–
$\exp(-M_1)$ - *SpinGlass*	9.86	–	0.71	–
$\exp(-M_2)$ - *SpinGlass*	14.8	–	0.63	–
$\exp(-M_\infty)$ - *SpinGlass*	31.86	–	0.58	–

Maximization tem resultados inferiores sem a informação do número de grupos.

Wine

Esta base de dados apresenta característica da análise química de vinhos, para que, a partir desta seja possível determinar sua origem (AEBERHARD; COOMANS; De Vel, 1992). Este conjunto de dados está dividido em três classes, sendo a primeira com 59 elementos, a segunda com 71 e a última com 49. A característica fundamental deste conjunto é ser "bem comportado", não sendo um problema muito desafiador do ponto de vista de classificação supervisionada. Os atributos que descrevem esta são: álcool, ácido málico, cinza, alcalinidade da cinza, magnésio, fenóis totais, flavonóides, fenóis não flavonóides, proantocianinas, intensidade de cor, Hue, matiz, OD280/OD315 de vinhos diluídos e Prolina.

Para esta base de dados todos os atributos foram normalizados na etapa de pré-processamento dos dados.

A avaliação realizada foi semelhante à feita para a base de dados Iris, ou seja, comparou-se a metodologia proposta aos métodos tradicionais, expostos no capitulo 2. Para a abordagem baseada em redes foram testadas várias combinações entre métodos de detecção de comuni-

dades e medidas de similaridade. Os melhores resultados são mostrados na tabela 4.2. Nesta são mostrados os valores do erro obtido, bem como o índice de Jaccard. Utilizou-se também a informação relativa ao número de grupos quando possível.

Tabela 4.2: Erros obtidos para o agrupamento na base de dados *Wine*, considerando os casos onde o número de *clusters* é conhecido, $k = 3$, ou não, $k =?$. Os algoritmos EM, k-médias e *Farthest First* utilizam esta informação. A abordagem baseada em redes foi testada para as medida de similaridade baseadas na exponencial da distância de Minkowski e três métodos diferentes para reconhecimento da estrutura de comunidades, *FastGreedy*, *LabelPropagation* e *SpinGlass*.

Method	% error ($k =?$)	% error ($k = 3$)	Jaccard ($k =?$)	Jaccard ($k = 3$)
k-means	–	5.61	–	0.80
cobweb	93.82	–	0.0078	–
farthest first	–	30.33	–	0.46
EM	19.66	2.80	0.63	0.89
$M_{0.5}$ - *FastGreedy*	1.68	1.68	0.93	0.93
M_1 - *FastGreedy*	1.12	1.12	0.95	0.954
M_2 - *FastGreedy*	21.91	21.91	0.51	0.51
M_∞ - *FastGreedy*	15.73	15.73	0.57	0.57
$M_{0.5}$ - *SpinGlass*	5.73	–	0.80	–
M_1 - *SpinGlass*	5.28	–	0.81	–
M_2 - *SpinGlass*	5.39	–	0.81	–
M_∞ - *SpinGlass*	8.9	–	0.71	–
$M_{0.5}$ - *LabelPropagation*	60.11	–	0.33	–
M_1 - *LabelPropagation*	60.11	–	0.33	–
M_2 - *LabelPropagation*	60.11	–	0.33	–
M_∞ - *LabelPropagation*	60.11	–	0.33	–

A análise da tabela 4.2 indica que a metodologia proposta apresenta resultados muito bons se comparados aos método tradicionais, porém, neste caso o desempenho do método de reconhecimento de comunidades baseado em propagação de rótulos apresentou um resultado bastante inferior se comparado aos demais métodos de reconhecimento de comunidades, porém seu índice de Jaccard é comparável ao do método *Farthest First*, mesmo apresentando uma grande diferença em relação ao erro. Quanto às demais configurações, o desempenho foi sempre preciso, mesmo quando não foi informado o número de *clusters*.

Quanto aos métodos tradicionais, o EM apresenta um resultado preciso quando tem a informação referente ao número de *clusters*, porém tem seu desempenho muito prejudicado sem esta. Isto não aconteceu quando utilizou-se a metodologia proposta com o método "$M_{0.5}$ -

FastGreedy", cujo erro foi 1.68539% e $J = 0.93186$, com ou sem a informação do número de grupos.

4.2 Conclusões

Pela análise dos resultados obtidos com as bases sintéticas, verifica-se que a abordagem baseada em redes se destaca tanto em definições de grupos baseadas em centróide, quanto em definições baseadas em continuidade. Sugerindo, assim, que este método pode ser aplicado à uma vasta gama de bases de dados reais. Os resultados obtidos para as bases de dados reais Iris e *Wine* também conduzem a conclusões semelhantes.

A metodologia apresentada neste trabalho se mostrou bastante robusta, obtendo bons resultados em grande parte dos experimentos executados. É importante entender que todo problema de agrupamento de dados estará sujeito a métrica escolhida para o espaço de atributos. Com a abordagem baseada em redes isto não é diferente. Foram propostas várias métricas possíveis, além da proposta e análise de uma técnica para limiarização que visa um ganho de desempenho, reduzindo o custo computacional. No entanto, foram analisados diversos métodos de reconhecimento de comunidades, possibilitando um grande número de combinações.

A abordagem proposta pode apresentar um custo computacional relativamente baixo se for utilizado, por exemplo, o método de propagação de rótulos. Caso a base de dados seja menor, mas não necessariamente pequena, é possível utilizar o método guloso, que apresentou resultados bastante precisos nos experimento realizados. Já o método baseado em mecânica estatística é o mais caro computacionalmente dentre os três considerados. Porém, tal método apresenta possibilidades interessantes que deverão ser avaliadas em trabalhos futuros, como por exemplo em agrupamento fuzzy.

Como trabalhos futuros, pretende-se verificar uma metodologia fuzzy baseada em redes complexas, bem como um agrupamento hierárquico, ou seja, sub-grupos dentro de grupos. Isto pode ser feito, a principio utilizando o algoritmo já utilizado neste trabalho (REICHARDT; BORNHOLDT, 2006) e posteriormente explorar outros algoritmos.

Referências Bibliográficas

AEBERHARD, S.; COOMANS, D.; De Vel, O. Comparison of classifiers in high dimensional settings. *Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep*, 1992.

ALBERT, R.; JEONG, H.; BARABÁSI, A. Internet: Diameter of the world-wide web. *Nature*, Nature Publishing Group, v. 401, n. 6749, p. 130–131, 1999.

ASUNCION, A.; NEWMAN, D. *UCI machine learning repository, 2007*. 2007.

BISHOP, C. *Pattern recognition and machine learning*. [S.l.]: Springer New York, 2006.

BOCCALETTI, S. et al. Complex networks: Structure and dynamics. *Physics reports*, Elsevier, v. 424, n. 4-5, p. 175–308, 2006. ISSN 0370-1573.

CLAUSET, A.; NEWMAN, M.; MOORE, C. Finding community structure in very large networks. *Physical review E*, APS, v. 70, n. 6, p. 066111, 2004.

CORTER, J.; GLUCK, M. Information, uncertainty, and the utility of categories. In: *Proceedings of the Seventh Annual Conference of the Cognitive Science Society. Lawrence Erlbaum Associates*. [S.l.: s.n.], 1985. p. 283–287.

DEMPSTER, A.; LAIRD, N.; RUBIN, D. Maximum likelihood from incomplete data via the em algorithm, jr statist. *Soc. B*, v. 39, n. 1, p. 1–38, 1977.

ERDOS, P.; RENYI, A. On random graphs. *Publicationes mathematicae*, v. 6, n. 290-297, p. 53–54, 1959.

FALOUTSOS, M.; FALOUTSOS, P.; FALOUTSOS, C. On power-law relationships of the intemet topology. *ACM SIGCOMM Computer Communication Review*, v. 29, n. 4, p. 251–262, 1999.

FISHER, D. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, Springer, v. 2, n. 2, p. 139–172, 1987.

FISHER, R. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936.

FORTUNATO, S. Community detection in graphs. *Physics Reports*, Elsevier, v. 486, n. 3-5, p. 75–174, 2010.

GIRVAN, M.; NEWMAN, M. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 99, n. 12, p. 7821, 2002.

HOCHBAUM, D.; SHMOYS, D. A best possible heuristic for the k-center problem. *Mathematics of operations research*, JSTOR, p. 180–184, 1985.

JAIN, A.; DUBES, R. *Algorithms for clustering data*. [S.l.]: Prentice-Hall, Inc., 1988.

KIRKPATRICK, S.; GELATT, C. MP Vecchi Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983.

LAARHOVEN, P.; LAARHOVEN, P. van; AARTS, E. *Simulated annealing: theory and applications*. [S.l.]: Springer, 1987. ISBN 9027725136.

MERVIS, C.; ROSCH, E. Categorization of natural objects. *Annual review of psychology*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 32, n. 1, p. 89–115, 1981.

NEWMAN; GIRVAN. Finding and evaluating community structure in networks. *Physical Review E*, v. 69(026113), p. 15, 2004.

NEWMAN, M. Analysis of weighted networks. *Physical Review E*, APS, v. 70, n. 5, p. 056131, 2004.

NEWMAN, M. Fast algorithm for detecting community structure in networks. *Physical Review E*, APS, v. 69, n. 6, p. 066133, 2004.

RAGHAVAN, U.; ALBERT, R.; KUMARA, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, APS, v. 76, n. 3, p. 036106, 2007.

REICHARDT, J.; BORNHOLDT, S. Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters*, APS, v. 93, n. 21, p. 218701, 2004.

REICHARDT, J.; BORNHOLDT, S. Statistical mechanics of community detection. *Physical Review E*, APS, v. 74, n. 1, p. 016110, 2006.

RODRIGUES, F.; ARRUDA, G. de; COSTA, L. A complex networks approach for data clustering. *Arxiv preprint arXiv:1101.5141*, 2011.

TAN, P. et al. *Introduction to data mining*. [S.l.]: Pearson Addison Wesley Boston, 2006.

THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern recognition*. [S.l.]: Academic Press, 2003.

VIDAL, V. *Applied simulated annealing*. [S.l.]: Springer, 1993. ISSN 0075-8442. ISBN 354056229X.

ZACHARY, W. An information flow model for conflict and fission in small groups, 1977. *J. Anthropol. Res*, v. 33, p. 452.