# Técnica de aprendizado semissupervisionado para detecção de outliers

***Fabio Willian Zamoner***

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:_____

# Técnica de aprendizado semissupervisionado para detecção de outliers

**Fabio Willian Zamoner**

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

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

**USP – São Carlos**
**Novembro de 2013**

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

ZZ25tat

Zamoner, Fabio Willian
Técnica de aprendizado semissupervisionado para detecção de outliers / Fabio Willian Zamoner; orientador Zhao Liang. -- São Carlos, 2013.
73 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2013.

1. detecção de outliers. 2. competição e cooperação de partículas. 3. aprendizado semissupervisionado. I. Liang, Zhao, orient. II. Título.

# Agradecimentos

Sem a presença de Deus em minha vida, eu não conseguiria superar os obstáculos. Agradeço ao Senhor pela proteção e pela minha saúde.

Minha imensa gratidão ao professor Zhao Liang por me orientar e por dar apoio integral ao desenvolvimento deste trabalho. Aconselhou-me não somente sobre o assunto abordado aqui como também sobre o modo que um pesquisador deveria agir. Acreditou no meu potencial mesmo sabendo de minhas limitações. Ter a oportunidade de trabalhar com você foi muito gratificante.

Aos meus amigos da pós-graduação por suas sugestões que contribuíram significativamente para o aperfeiçoamento do conteúdo deste trabalho.

Ao CNPq e à FAPESP pelo apoio financeiro.

Aos meus amigos de república Biro, Cidão, Eduardo, Febem, Job, Paulista, Pincel, Raul e Tiago pela amizade e convivência harmoniosa nos anos que morei em São Carlos. Agradeço também aos meus amigos André, Michel e Leonardo pelos momentos de alegria e distração. Vocês serviram de inspiração para minha carreira profissional.

Um obrigado especial ao meu tio Paulo pelas caronas oferecidas e aos meus padrinhos pelo incentivo e ajuda grandiosa nos momentos difíceis da vida.

À toda minha família, em especial, ao meu pai, à minha mãe e à minha vó Thereza pelo amor e carinho. Vocês sempre me deram a motivação e o apoio necessário para alcançar meus objetivos.

# Resumo

Detecção de *outliers* desempenha um importante papel para descoberta de conhecimento em grandes bases de dados. O estudo é motivado por inúmeras aplicações reais tais como fraudes de cartões de crédito, detecção de falhas em componentes industriais, intrusão em redes de computadores, aprovação de empréstimos e monitoramento de condições médicas. Um *outlier* é definido como uma observação que desvia das outras observações em relação a uma medida e exerce considerável influência na análise de dados. Embora existam inúmeras técnicas de aprendizado de máquina para tratar desse problema, a maioria delas não faz uso de conhecimento prévio sobre os dados. Técnicas de aprendizado semissupervisionado para detecção de *outliers* são relativamente novas e incluem apenas um pequeno número de rótulos da classe normal para construir um classificador. Recentemente um modelo semissupervisionado baseado em rede foi proposto para classificação de dados empregando um mecanismo de competição e cooperação de partículas. As partículas são responsáveis pela propagação dos rótulos para toda a rede. Neste trabalho, o modelo foi adaptado a fim de detectar *outliers* através da definição de um escore de *outlier* baseado na frequência de visitas. O número de visitas recebido por um *outlier* é significativamente diferente dos demais objetos de mesma classe. Essa abordagem leva a uma maneira não tradicional de tratar os *outliers*. Avaliações empíricas sobre bases artificiais e reais demonstram que a técnica proposta funciona bem para bases desbalanceadas e atinge precisão comparável às obtidas pelas técnicas tradicionais de detecção de *outliers*. Além disso, a técnica pode fornecer novas perspectivas sobre como diferenciar objetos, pois considera não somente a distância física, mas também a formação de padrão dos dados.

# Abstract

Outlier detection plays an important role for discovering knowledge in large data sets. The study is motivated by a plethora of real applications such as credit card frauds, fault detection in industrial components, network intrusion detection, loan application processing and medical condition monitoring. An outlier is defined as an observation that deviates from other observations with respect to a measure and exerts a substantial influence on data analysis. Although numerous machine learning techniques have been developed for attacking this problem, most of them work with no prior knowledge of the data. Semi-supervised outlier detection techniques are relatively new and include only a few labels of normal class for building a classifier. Recently, a network-based semi-supervised model was proposed for data classification by employing a mechanism based on particle competition and cooperation. Such particles are responsible for label propagation throughout the network. In this work, we adapt this model by defining a new outlier score based on visit frequency counting. The number of visits received by an outlier is significantly different from the remaining objects. This approach leads to an unorthodox way to deal with outliers. Our empirical evaluations on both real and simulated data sets demonstrate that the proposed technique works well with unbalanced data sets and achieves a precision compared to traditional outlier detection techniques. Moreover, the technique might provide new insights into how to differentiate objects because it considers not only the physical distance but also the pattern formation of the data.

# Sumário

# Lista de Figuras

# Lista de Tabelas

Capítulo 1

# Introdução

Encontrar padrões em conjunto de dados é uma importante tarefa na mineração de dados. Atualmente, as empresas trabalham com grande volume de dados e necessitam de técnicas robustas para aquisição de conhecimento. Analisar manualmente as bases de dados tornou-se uma tarefa impraticável, por isso ferramentas para análise automática têm sido desenvolvidas.

Uma das áreas fortemente relacionada à mineração de dados e que evoluiu muito nos últimos anos é a área de aprendizado de máquina. A evolução se deu pela variedade de métodos eficientes capazes de manipular grande quantidade de dados e também fornecer ferramentas para tratamento dos dados e aquisição de conhecimento. Mitchell (1997) define aprendizado de máquina como a capacidade de melhorar o desempenho na realização de uma certa tarefa por meio da experiência. Esse aprendizado possibilitou o desenvolvimento de métodos mais robustos em relação aos meios convencionais conhecidos como sistemas especialistas ou sistemas baseados em conhecimento os quais são programados por um especialista com grande conhecimento sobre um certo problema. Os métodos de aprendizado de máquina são mais autônomos diminuindo a necessidade de intervenção humana destacando-se em aplicações como reconhecimento de caracteres, inteligência artificial em jogos, detecção de fraudes de cartões de crédito, diagnóstico de doenças e predição de taxa de cura de pacientes, pesquisa de mercado para obter informações sobre quais produtos são comprados em conjunto, segmentação de imagens para detectar bordas de objetos e ruídos, entre outros (Alpaydin, 2010; Faceli et al., 2011; Gan et al., 2007).

Os paradigmas mais comuns de aprendizado de máquina são: supervisionado e não supervisionado. No aprendizado supervisionado, o conhecimento pode ser adquirido com auxílio de uma função a qual é estimada em uma fase de treinamento de um modelo usando dados previamente rotulados por especialistas. Com a função estimada, é possível inferir os rótulos

desconhecidos de outros dados. Quando a forma de aquisição de conhecimento ocorre sem considerar rótulos dos dados de entrada, por exemplo, por meio de agrupamento de dados, o tipo de aprendizado é chamado de não supervisionado. Neste caso, o objetivo é encontrar padrões formados pelos dados. Os métodos de agrupamento de dados são usados principalmente para fins de sumarização e compressão de dados, identificação de vizinhos mais próximos de forma mais eficiente que outras técnicas e descobrimento de estruturas escondidas (Tan et al., 2005).

Outro paradigma de aprendizado de máquina é chamado de aprendizado semissupervisionado. Ele é visto como um paradigma híbrido que está entre o aprendizado supervisionado e o não supervisionado, pois utiliza tanto os dados não rotulados como também alguns dados previamente rotulados. Têm alto valor prático na indústria, pois uma pequena quantia de exemplos rotulados é naturalmente conhecida sendo aplicado em diversos problemas reais como: agrupamento de dados de expressão genética que tenham padrões similares, categorização de textos e reconhecimento de fala (Zhu, 2005).

Um tópico importante da mineração de dados é a detecção de *outliers*. O objetivo é identificar dados que estão fora do padrão e que desviam acentuadamente em relação aos demais. Existem diversas definições para o termo *outlier* (D. M. Hawkins, 1980; Barnett e Lewis, 1995; Grubbs, 1969) e as causas mais frequentes de sua aparição são erros na medição ou execução que provocam alterações significativas nas análises de toda a base de dados. Detectar dados que apresentam um comportamento diferente dos demais é importante em aplicações como processamento de imagens (Singh e Markou, 2004), dados biológicos (P. Sun et al., 2006), fraudes de cartão de crédito (Kou et al., 2004), detecção de intrusão (Portnoy et al., 2001), monitoramento de tráfego (Shekhar et al., 2001), distúrbios em ecossistemas (Kou e Lu, 2006) e análise de emails (Shetty e Adibi, 2005). Em razão disso, várias revisões bibliográficas foram feitas (Chandola et al., 2009; Hodge e Austin, 2004; Agyemang et al., 2006; Su e Tsai, 2011). Contudo, os *outliers* são, na maioria das vezes, eventos raros e se parecem com dados normais tornando o problema de detecção difícil de ser tratado.

Grafo [1] é uma poderosa forma de representação dos dados exibindo com mais facilidade as relações entre os dados se comparado à representação na forma atributo-valor. Ele pode além de facilitar a visualização dos dados, fornecer informações para identificação de padrões e estruturas complexas. Grafos são amplamente estudados principalmente pelas áreas de ciências de computação e sociologia (Newman, 2004). O interesse veio da capacidade na modelagem de sistemas complexos reais representando a dinâmica e funções deles. Comunidade é um exemplo de estrutura complexa comumente encontrada em redes sociais, redes biológicas, na *World Wide Web* e na Internet (Fortunato, 2010). Ela é caracterizada por grupos de vértices densamente conectados com poucas conexões entre vértices de comunidades diferentes. Além de favorecer o estudo dos mecanismos de crescimento e formação da rede (Clauset, 2005; Newman e Girvan, 2004), a identificação de comunidades pode revelar os sites que tratam de tópicos relacionados em redes de web sites, interesses em comum de indivíduos que pertencem a uma mesma comunidade em redes sociais, funções similares de elementos de mesma comunidade

[1] Neste trabalho, grafo e rede são intercambiáveis

em redes de circuito eletrônico e em uma rede neural, etc. Consequentemente, tornou-se um tópico importante para mineração de dados.

Problemas de agrupamento e classificação de dados podem ser tratados com técnicas baseadas em grafos. No primeiro caso, a tarefa de detecção de comunidades pode ser realizada com técnicas que apoiam-se na teoria espectral de redes (Pothen et al., 1990), modelo de *Potts* (Reichardt e Bornholdt, 2004), medida *betweeness* (Newman e Girvan, 2004), caminhada aleatória (Zhou, 2003), entre outras. Já as tarefas de classificação são usualmente realizadas com técnicas de propagação de rótulos (Chapelle et al., 2006). Embora as técnicas baseadas em grafos sejam bastante estudadas, a aplicação delas na detecção de *outliers* é recente (Berton et al., 2010; Noble e Cook, 2003; Moonesignhe e Tan, 2006; Costa et al., 2009; Hautamaki et al., 2004; Chen et al., 2011). Para esse tipo de problema, tais técnicas geralmente são desenvolvidas com objetivo de identificar vértices que apresentem comportamento diferenciado em relação aos vértices vizinhos. No contexto de redes de computadores, elas estão incluídas em sistemas de análise de tráfego e prevenção de falhas para identificar máquinas que são alvo de ataques ou que se comportam de maneira suspeita. Um dos grandes desafios é conseguir diferenciar vértices *outliers* dos vértices normais já que a diferença entre eles nem sempre é evidente.

Recentemente, foram desenvolvidos modelos dinâmicos que se baseiam em um novo tipo de aprendizado competitivo (Breve et al., 2011; Silva e Zhao, 2012; Quiles et al., 2008). A ideia por trás desses modelos é a utilização de partículas que caminham sobre a rede que representa uma base dados. Particularmente, o modelo semissupervisionado de Breve et al. (2011) realiza a propagação de rótulos através da competição e cooperação de times de partículas. A competição ocorre entre times de partículas que disputam por vértices e a cooperação ocorre entre partículas do mesmo time que marcam o território por onde passam a fim de impedir que partículas intrusas tomem posse. Em cada instante de tempo, o movimento de cada partícula pode ser aleatório ou preferencial. Quando um movimento preferencial é realizado, ela visita os vértices que já estão dominados por seu time com intuito de proteger o território. No movimento aleatório, a partícula tem comportamento exploratório visando conquistar territórios ainda não explorados. Ao final do processo, cada time de partículas possui um conjunto de vértices dominados os quais serão rotulados de acordo com a classe do time de partículas. Por apresentar bons resultados na classificação de dados, esse mecanismo vem sendo explorado por diversos pesquisadores.

## 1.1 Objetivos

Os principais objetivos deste trabalho estão listados a seguir.

- O tópico de detecção de *outliers* é tipicamente visto como um problema de cálculo de densidade. Entretanto, abordagens não baseadas em densidade têm obtido resultados interessantes neste contexto indicando que soluções alternativas também são adequadas para tratar esse problema. Um dos objetivos deste trabalho é identificar *outliers* pela formação de padrão dos dados, não somente pela característica física;

- O mecanismo de competição e cooperação de partículas tem sido aplicado em problemas de classificação (Breve et al., 2011; Silva e Zhao, 2012) e agrupamentos de dados (Quiles et al., 2008). Neste trabalho, deseja-se investigar informações fornecidas por esse mecanismo para caracterização de vértices. Algumas informações facilmente obtidas são: número de visitas, potencial da partícula e número de vezes que um vértice trocou de dono;
- Propor uma técnica para detecção de *outliers* baseada na frequência de visitas. Ela consiste de uma medida que atribua grau de anormalidade para cada dado segundo a perspectiva da competição e cooperação de partículas;

Maiores detalhes do projeto são apresentados no Capítulo 4.

## 1.2 Motivação

Técnicas tradicionais de detecção de *outliers* normalmente seguem o paradigma não supervisionado, ou seja, não fazem uso de rótulos da classe normal ou de *outliers*. Todavia, é muito comum o conhecimento prévio de alguns rótulos da classe normal. Tais rótulos podem ser úteis na detecção daqueles dados já que auxiliam o modelo a distinguir entre o comportamento normal e anormal. Portanto, faz-se necessário investigar abordagens semissupervisionadas tanto pelas inúmeras aplicações práticas quanto pelo benefício da disponibilidade de rótulos.

No contexto de detecção de *outliers*, a utilização de técnicas baseadas em rede é recente. *Outliers* são geralmente estudados no campo da estatística seja por meio de medidas relacionadas as distribuições dos dados ou por meio do cálculo de densidade. Por outro lado, técnicas baseadas em rede dispõem de medidas mais robustas que podem ajudar a identificar padrões e, consequentemente, detectar *outliers*. Tais medidas são divididas em locais, intermediárias e globais. As locais estão relacionadas a um único vértice ou aresta, as intermediárias estão relacionadas com grupos de vértices e as globais relacionadas com caminhos ou circuitos. Diante do exposto, redes fornecem uma ampla variedade de medidas sob diferentes perspectivas para diferenciar os dados.

O mecanismo de competição e cooperação de partículas mostrou-se promissor na realização de tarefas de detecção de comunidades (Quiles et al., 2008) e classificação de dados (Breve et al., 2011; Silva e Zhao, 2012). No primeiro caso, observou-se que este mecanismo tem a capacidade de identificar estruturas sobrepostas. Em redes modulares, vértices sobrepostos são aqueles que estão localizados na fronteira entre comunidades. Esses vértices são difíceis de classificar e na prática podem pertencer a mais de uma comunidade ou *cluster*. Em redes sociais, na qual os indivíduos são representados por vértices e as relações representadas por arestas, frequentemente indivíduos possuem amizades com pessoas de diferentes comunidades como família, escola, etc. Muitos algoritmos falham na identificação de comunidades sobrepostas. Já em tarefas de classificação de dados, os modelos firmados neste mecanismo, descritos em (Breve et al., 2011; Silva e Zhao, 2012), obtiveram desempenho superior a vários outros modelos baseados em redes mostrando habilidade para classificar dados que formam estruturas não

convencionais. Além dessas vantagens, considerando uma rede esparsa e já está construída, eles possuem baixo custo computacional com ordem de complexidade linear em relação ao número de vértices.

Embora o modelo de competição e cooperação de partículas tenha sido aplicado em problemas tradicionais de aprendizado de máquina, ainda não há uma abordagem para tratar de *outliers*. O mecanismo mencionado anteriormente fornece valiosas informações para caracterização de vértices. Algumas delas são facilmente obtidas e incluem o número de visitas que um vértice recebeu, valor final do potencial do vértice que caracteriza o nível de dominação e o número de vezes que o vértice trocou de dono, etc. Logo, a detecção de *outliers* pode ser realizada por meio de medidas de caracterização de vértices.

## 1.3 Organização do documento

Este documento está organizado da seguinte forma. O Capítulo 2 aborda os três principais paradigmas de aprendizado de máquina: supervisionado, não supervisionado e semissupervisionado. Neste último paradigma, é descrito um modelo de competição e cooperação de partículas no qual a técnica proposta é fundamentada. O Capítulo 3 é dedicado à detecção de *outliers*. Ele inclui as principais definições de *outlier*, uma discussão sobre a importância desse assunto em aplicações industriais e categorias de técnicas de detecção usualmente citadas na literatura. No Capítulo 4, uma técnica para detecção de *outliers* é apresentada. Também são apresentados os resultados obtidos em bases de dados reais e artificiais, além da análise de um parâmetro proposto da técnica. Por último, as conclusões e os trabalhos futuros são discutidos no Capítulo 5.

Capítulo 2

# Aprendizado de Máquina

Aprendizado de máquina é uma sub-área da Inteligência Artificial e envolve pesquisa de técnicas e algoritmos que melhoram seu desempenho com a experiência adquirida (Alpaydin, 2010). Deseja-se que o computador seja capaz de induzir automaticamente o algoritmo durante o aprendizado a partir de exemplos de dados. Basicamente, busca-se um classificador que melhor se adapta ao conjunto de dados do problema (Mitchell, 1997). Conforme Mitchell (1997), é conveniente determinar o tamanho ou complexidade do espaço do classificador, a precisão aproximada para o tal problema, a probabilidade de obter uma classificação aceitável e a maneira como os conjuntos de treinamento serão apresentados.

Aprendizado de máquina está presente em vários processos. A mineração de dados, por exemplo, utiliza algoritmos de aprendizagem com objetivo de extrair conhecimento de grandes bases de dados (Alpaydin, 2010). Esse conhecimento ajuda a entender o problema e permite fazer previsões de resultados. Aplicações de reconhecimento de padrões também incluem o aprendizado de máquina no processo de identificação de faces, de fala e caracteres manuscritos, diagnósticos médicos, biometria, extração de conhecimento, identificação de exceções dos dados de entrada, etc.

Os principais paradigmas do aprendizado de máquina são: supervisionado e não-supervisionado. No aprendizado supervisionado o objetivo é construir, a partir de exemplos rotulados, um classificador que seja capaz de predizer o rótulo de exemplos ainda não vistos. Os exemplos previamente rotulados são geralmente fornecidos por um especialista e usados durante a fase de treinamento para induzir uma função que retorne o rótulo para um exemplo de entrada. Quando a saída da função contém apenas valores discretos o problema é chamado de classificação, caso contrário, é chamado de regressão. No aprendizado não-supervisionado os rótulos das classes de padrões de treinamento são ignorados ou estão indisponíveis, e o interesse está na organização dos padrões de entrada (Theodoridis e Koutroumbas, 2008). Esse tipo de

aprendizado pode ser realizado por meio de agrupamento ou clusterização de dados em que os dados são divididos em grupos de maneira que dados similares pertençam ao mesmo grupo ou *cluster*.

Em aplicações reais, apenas uma quantia pequena de dados está previamente rotulada sendo insuficiente para gerar um modelo confiável. Como a tarefa de rotulação manual dos dados por especialistas é custosa, foi preciso buscar novas alternativas para a realização do aprendizado. Uma delas é por meio da combinação de uma quantia pequena disponível de dados rotulados com a grande maioria de dados não rotulados. Métodos que seguem este paradigma pertencem ao aprendizado semissupervisionado.

## 2.1 Conceitos Gerais

Nesta seção são descritos os conceitos fundamentais para os conteúdos abordados nas próximas seções.

### 2.1.1 Tipos de atributos

Define-se atributo como uma propriedade ou característica de um objeto e ele pode ser dividido em quatro tipos:

**nominal:** atributos cujos valores são associados a uma qualidade ou categoria. Não existe uma ordem entre os valores e somente fornecem informações suficientes para distinguir um valor de outro (Tan et al., 2005). São exemplos de atributos nominais: CEP, cor dos olhos (castanhos, verdes, azuis) e gênero.

**ordinal:** Similar ao anterior exceto pelo fato de que esse atributo possui valores que podem ser ordenados. Exemplos: número de ruas, hierarquia militar, dureza de materiais, etc.

**intervalar:** além da relação de ordem, esses atributos aceitam operações de adição e subtração. Temperatura na escala Célsius e datas de um calendário são exemplos de atributos intervalares.

**racional:** tipo mais completo que permite operações de multiplicação e divisão além das operações descritas dos outros tipos. Exemplos: idade, temperatura na escala Kelvin, altura, salário de um empregado, etc.

Enquanto os dois primeiros tipos de atributos são classificados como categóricos ou qualitativos, os tipos intervalar e racional são classificados como numéricos ou quantitativos. Os atributos possuem outras características importantes como a escala, quantidade de valores que podem assumir, etc.

### 2.1.2 Medidas de similaridade e dissimilaridade

Medidas de proximidade são utilizadas para medir a força da relação entre dados. Uma medida de proximidade pode representar a dissimilaridade se o valor é proporcional à distância entre os elementos ou representar similaridade caso contrário. As medidas que computam a proximidade entre dados variam de acordo como o tipo de atributo que suportam sendo que a maioria das medidas é destinada para atributos numéricos. O conjunto de todos os pares de proximidade entre dados forma a matriz de proximidade que geralmente é simétrica. Toda medida de similaridade e de dissimilaridade é simétrica e não negativa (Xu e Wunsch, 2005). Se satisfizer as condições de não negativa, reflexiva, comutativa e de desigualdade triangular é chamada de métrica (Gan et al., 2007).

Seja um conjunto de dados $D$ com $n$ dados. A medida definida pela equação 2.1 é chamada de *distância de Minkowski*. Atribuindo $r = 1$ obtém-se a *distância de Manhattan*, também denominada de *distância bloco-cidade* e com $r = 2$ obtém-se a *distância Euclidiana*, uma das medidas de dissimilaridade mais utilizadas.

$$D(x_i, x_j) = (\sum_{l=1}^{d} |x_{il} - y_{jl}|^r)^{\frac{1}{r}}, \qquad r \geq 1. \tag{2.1}$$

Outra medida bem conhecida é a *distância de Mahalanobis* definida pela equação 2.2 em que $\sum$ é a matriz de covariância de ordem $d \times d$. A matriz de covariância (equação 2.3) é simétrica sendo obtida por meio da matriz $X$ de ordem $n \times d$ onde cada linha representa um dado. Enquanto a diagonal da matriz $\sum$ representa a variância de cada atributo, os elementos fora da diagonal medem a redundância entre pares de atributos. A *distância de Mahalanobis* é invariante a todas as transformações não singulares e pode aliviar distorções causadas pela combinação linear de atributos (Gan et al., 2007). A *distância Euclidiana* é um caso especial dessa distância quando a distribuição dos dados é uniforme, isto é, quando as características não estão correlacionadas (Xu e Wunsch, 2005).

$$D(x_i, x_j) = \sqrt{(x_i - x_j)\sum^{-1}(x_i - x_j)^T}, \tag{2.2}$$

$$\sum = \frac{1}{n} X^T X \tag{2.3}$$

O *coeficiente de correlação de Pearson* é uma medida de similaridade amplamente usada em bases de dados genéticos e em processamento de imagens. Entretanto, pode não ser uma boa escolha quando bases de dados contêm *outliers* ou quando os dados seguem uma distribuição não *Gaussiana* (Jiang et al., 2004). Para dois dados $x_i$ e $x_j$ com médias iguais a $\bar{x_i}$ e $\bar{x_j}$ respectivamente, o coeficiente é definido como:

$$r_{pearson}(x_i, x_j) = \frac{\sum_{l=1}^{d}(x_{il} - \bar{x_i})(x_{jl} - \bar{x_j})}{\sqrt{\sum_{l=1}^{d}(x_{il} - \bar{x_i})^2}\sqrt{\sum_{l=1}^{d}(x_{jl} - \bar{x_j})^2}} \tag{2.4}$$

Quando o valor é próximo de 0 indica que os dados não tem relação linear. Valor positivo indica que os dados são diretamente correlacionados e valor negativo indica que os dados são inversamente correlacionados. A distância é dada por $D(x_i, x_j) = 1 - |r_{pearson}(x_i, x_j)|$.

A similaridade do cosseno (equação 2.5), onde $\langle x_i, x_j \rangle$ indica o produto interno e $\| \cdot \|$ a norma do vetor , é bastante usada para dados de transações, os quais ocorrem em aplicações como cestas de compras, recuperação de informação e mineração na Web combinando regras de associação e agrupamento de dados para aquisição de conhecimento.

$$cos(x_i, x_j) = \frac{\langle x_i, x_j \rangle}{\|x_i\| \cdot \|x_j\|} \tag{2.5}$$

Para medir a similaridade entre *clusters* sob o ponto de vista estatístico pode-se utilizar a medida *Kullback-Leibler Divergence* (*KLD*). Essa medida é definida pela entropia relativa de duas funções de densidade de probabilidade conforme a equação 2.6. Se as duas distribuições de probabilidade são idênticas, então o valor é zero. Por convenção $0 \log(\frac{0}{q}) = 0$ e $p \log(\frac{p}{0}) = \infty$.

$$KLD(x_i, x_j) = \frac{1}{2}(\sum_{l=1}^{d} x_{il} \log \frac{x_{il}}{x_{jl}} + \sum_{l=1}^{d} x_{jl} \log \frac{x_{jl}}{x_{il}}) \tag{2.6}$$

A escolha de uma medida adequada exige o conhecimento do conjunto de dados. Vale lembrar ainda que algumas métricas necessitam da normalização dos atributos visto que os atributos de maior escala tendem a dominar os demais. Outro detalhe importante diz respeito à maldição da dimensionalidade. Esse problema refere-se à dificuldade em determinar relações de proximidade entre os dados que possuem muitos atributos, pois o volume cresce exponencialmente com o número de dimensões. Quando a dimensão de um conjunto de dados é alta, torna-se menos significativa a diferença entre os dados mais próximos e os dados mais distantes. Dentre as formas de contornar esse problema destacam-se a atribuição de pesos maiores aos atributos mais relevantes e seleção ou combinação de atributos (redução de dimensionalidade).

## 2.2 Aprendizado supervisionado

Métodos de aprendizado supervisionado têm por objetivo a construção de um modelo que atribui um rótulo a um dado não rotulado. Seja $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ uma base de dados $d$-dimensional com $n$ objetos. Cada $x_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$ é um dado composto de $d$ valores tal que cada elemento $x_{ij}$ representa o valor atributo da dimensão $j$. A base de dados $\mathcal{X}$ pode ser dividida em dois conjuntos: o conjunto de treinamento $\mathcal{X}_{train}$ e o conjunto de teste $\mathcal{X}_{test}$. O primeiro conjunto é formado pelos dados cujos rótulos são conhecidos e pertencem ao conjunto $\mathcal{Y}_{train}$. Então, a tarefa é prever os rótulos desconhecidos $\mathcal{Y}_{teste}$ dos dados de teste $\mathcal{X}_{test}$.

Um modelo é chamado de classificador quando o valor (rótulo) retornado pertence a um conjunto discreto de valores ou de regressor quando o valor pertence a um conjunto infinito e ordenado de valores (Faceli et al., 2011). Em problemas de classificação, o modelo gera fronteiras de decisão que separam dados de uma classe das demais. Exemplo de uma fronteira de decisão em um problema de duas classes é a reta pontilhada desenhada na Figura 2.1. Tal reta é uma função cujos parâmetros são estimados usando os exemplos, ilustrados por quadrados e triângulos, previamente rotulados das duas classes $\mathcal{X}_{train}$. Ela servirá como hipótese para prever os rótulos $\mathcal{Y}_{teste}$ dos exemplos, ilustrados por círculos, pertencentes ao conjunto $\mathcal{X}_{test}$.

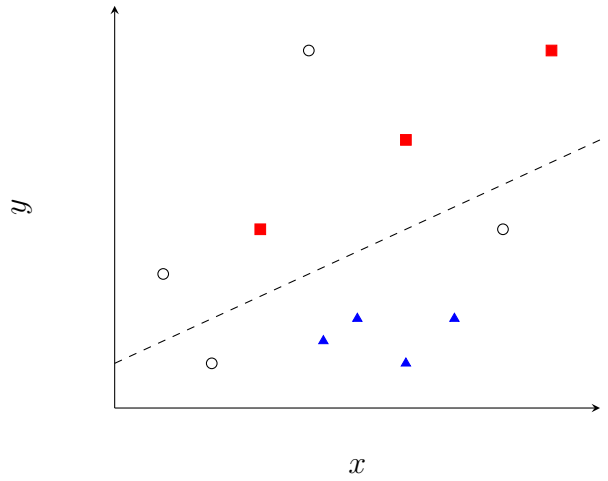



**Figura 2.1:** Exemplo de um problema de classificação binária na qual as classes estão separadas por uma reta pontilhada induzida com auxílio dos exemplos previamente rotulados. Cada forma geométrica corresponde a uma classe diferente exceto os círculos que correspondem aos dados não rotulados.

Três fases compõe o processo de estimação de um modelo: treinamento, validação e teste. Na fase de treinamento são estimados os parâmetros do modelo com o auxílio de um conjunto de dados previamente rotulados por especialistas. Então, o modelo obtido é analisado na fase de validação para verificar a capacidade de generalização. Finalmente, a fase de teste serve para verificar a confiabilidade do modelo através dos critérios de avaliação escolhidos obtendo o erro cometido ao classificar os dados de teste. Feitas tais considerações, duas possíveis situações devem ser destacadas: *overfitting* e *underfitting*. A primeira delas ocorre quando o modelo fica super ajustado aos dados de treinamento apresentando uma capacidade de generalização baixa para dados futuros os quais não fazem parte do conjunto de treinamento (Alpaydin, 2010). Já o termo *underfitting* refere-se à situação quando o modelo não apresenta bons resultados de classificação mesmo para os dados do conjunto de treinamento, isto é, o aprendizado foi abaixo do esperado. Pode-se concluir com essas duas situações que a geração de um modelo confiável requer não somente um conjunto de treinamento representativo como também critérios de avaliação bem definidos, preparação dos dados, etc.

Em um problema simples de duas classes, como aquele ilustrado na Figura 2.1, os exemplos que têm sido classificados pelo modelo podem ser separados em (Faceli et al., 2011):

**VP:** exemplos classificados com positivos e realmente são da classe positiva.

**VN:** exemplos classificados com negativos e realmente são da classe negativa.

**FP:** exemplos classificados com positivos, mas pertencem a classe negativa.

**FN:** exemplos classificados com negativos, mas pertencem a classe positiva.

Com esses dados em mãos, os modelos podem ser avaliados usando medidas como acurácia, medida-f, revocação, especificidade e precisão (Faceli et al., 2011). A precisão (*prec*) descrita na expressão 2.7 avalia a proporção dos dados da classe positiva preditos corretamente pelo classificador *f*. Tanto essa medida como outras medidas de avaliação descritas em (Faceli et al., 2011) podem ser facilmente modificadas para avaliar problemas multi-classe que são comumente encontrados no mundo real.

$$prec(f) = \frac{VP}{VP + FP} \tag{2.7}$$

## 2.3 Aprendizado não supervisionado

Ao contrário do aprendizado supervisionado, no paradigma não supervisionado as classes (ou rótulos) dos dados não são usados ou não estão disponíveis. O objetivo é identificar padrões ou estruturas formadas pelos dados. Uma categoria bem conhecida de métodos não supervisionados é chamada de agrupamento de dados. Agrupamento de dados (ou clusterização de dados) buscar dividir um conjunto de dados em *clusters* (ou grupos) tal que os dados de um mesmo *cluster* sejam mais similares que os dados de *clusters* diferentes considerando uma determinada medida de similaridade (Jain e Dubes, 1988).

### 2.3.1 Tipos de clusters

Apesar de não existir uma definição única para o conceito de *cluster*, ele pode ser entendido como uma coleção de dados que possuem alguma relação de proximidade. Os tipos de *cluster* são comumente categorizados em:

**bem separados:** ocorre quando qualquer dado de um *cluster* está mais próximo de cada um dos demais dados desse mesmo *cluster* que de outro *cluster*.

**baseados no centro:** os dados estão mais próximos do centro de seu *cluster* que dos centros dos demais *clusters*. Em geral, o centro de um *cluster* é um centróide definido pela média de todos os dados do *cluster* ou é um medóide definido pelo dado mais central do *cluster*.

**contínuos:** um dado está mais próximo de pelo menos outro dado do mesmo *cluster* que dos dados dos outros *clusters*.

**baseados em densidade:** *clusters* são regiões de alta densidade de pontos separados por regiões de baixa densidade.

**conceituais:** os dados de um mesmo *cluster* compartilham alguma característica em comum.

**descritos por uma função objetivo:** maximização ou minimização de uma função objetivo para gerar os *clusters*. O espaço de soluções tende a ser enorme sendo necessário avaliar a qualidade de um número restrito de soluções.

O aprendizado não supervisionado é um assunto bastante explorado pela comunidade científica sendo um componente essencial da mineração de dados e de aplicações de reconhecimento de padrões (Gan et al., 2007). Também são aplicados para representação sumarizada de grande volume de dados desde que o conjunto de dados tenha tendência a formar *clusters*. Geralmente o número de *clusters* é desconhecido e muitos algoritmos exigem o número de *clusters* como parâmetro.

### 2.3.2 Etapas do agrupamento de dados

O processo de agrupamento inclui as seguintes etapas (Jain e Dubes, 1988):

1. *Representação dos padrões:* consiste em representar os dados de forma adequada para ser usada por um algoritmo de agrupamento. A preparação dos dados pode melhorar o desempenho do algoritmo, a qualidade dos dados e facilitar a compreensão dos dados. Tal prática é muito importante, sobretudo quando a base de dados contém *outliers*, ruídos, dados duplicados, atributos inconsistentes, irrelevantes e incompletos.

2. *Definição de uma medida de proximidade:* necessária para medir a similaridade ou distância entre dois itens de dados. As medidas diferem principalmente no tipo de dado suportado sendo que a maioria das medidas foi feita para dados numéricos. Uma visão geral dessas medidas é apresentada na seção 2.1.2.

3. *Agrupamento:* etapa de execução do algoritmo de agrupamento. Uma variedade de tipos de algoritmos de agrupamento de dados são discutidos na seção 2.3.4.

4. *Abstração dos dados:* etapa não obrigatória na qual deseja-se obter uma representação compacta do conjunto de dados para fins de processamentos futuros com eficiência ou para facilitar a compreensão do conhecimento adquirido (Jain et al., 1999).

5. *Avaliação dos resultados:* é conveniente avaliar a confiabilidade do conhecimento extraído por meio de critérios de validação. Na seção 2.3.3 é feita uma breve descrição dos tipos de critérios.

Vários fatores devem ser considerados na escolha de um algoritmo de agrupamento. Cada algoritmo tem uma maneira particular de encontrar *clusters* em diferentes níveis de refinamento, isto é, variam na quantidade de *clusters* encontrados. A Figura 2.2 ilustra a divisão de um conjunto de dados em diferentes níveis de refinamento. Todos os resultados são válidos uma vez que a similaridade entre os dados é difícil de ser definida. Dividir um conjunto de dados em

muitos *clusters* dificulta a análise e interpretação dos resultados enquanto a geração de poucos *clusters* pode provocar perda de informação (Xu e Wunsch, 2005). O resultado ótimo, de acordo com a aplicação, é aquele que melhor representa as partições do conjunto de dados (Halkidi et al., 2002b).

### 2.3.3 Validação

A fim de selecionar o nível de refinamento mais adequado para a base de dados é preciso avaliar a qualidade do agrupamento por meio da validação de *clusters* (Halkidi et al., 2002b). Além de fornecer informações sobre a qualidade dos *clusters*, a validação também serve para determinar o número adequado de *clusters* da base de dados de acordo com um critério escolhido e para comparar resultados gerados por diferentes algoritmos de agrupamento. Para este fim, os seguintes critérios são empregados:

**Critérios internos:** utilizam apenas as quantidades e características dos dados originais como a matriz de similaridade. A forma de aplicação desses critérios depende da estrutura do agrupamento como no caso de algoritmos particionais e hierárquicos. Coesão de *clusters*, Separação de *clusters* e Silhueta são exemplos de critérios internos (Halkidi et al., 2002b).

**Critérios externos:** esse tipo de critério faz uso de uma estrutura pré-estabelecida imposta geralmente por um especialista. Índices mais famosos: *Rand*, *Jaccard*, *Hubert normalizado* e o *Fowlkes e Mallows* (Halkidi et al., 2002b; Faceli et al., 2011).

**Critérios relativos:** usados na estimação dos parâmetros do algoritmo de agrupamento que levam a melhores resultados. Também são usados na comparação entre resultados obtidos por diferentes algoritmos de agrupamento. Exemplos: família de índices Dunn e o índice Davies-Bouldin (Halkidi et al., 2002a).

Tanto os critérios internos quanto os externos apoiam-se em testes estatísticos e possuem alto custo computacional (Halkidi et al., 2002b). Por causa disso, as técnicas de Monte Carlo e de *bootstrap* são usualmente empregadas para reduzir o tempo computacional. A forma de validação mais adequada depende do conhecimento prévio do conjunto de dados (Faceli et al., 2011). Alguns critérios funcionam melhor para certos tipos de *clusters*. Determinar, por exemplo, o número correto de *clusters* de uma base de dados não é uma tarefa trivial.

Outra maneira é validar os *clusters* mediante teste de hipóteses, o qual fornece uma metodologia para comprovar se a hipótese assumida é verdadeira após análise das observações obtidas (Morettin e Bussab, 2003). É importante salientar que a comparação entre algoritmos de agrupamento pode não fazer sentido porque os resultados podem apresentar níveis de refinamento distintos. A interpretação dos *clusters* é realizada após a validação e geralmente por um especialista.

Outros fatores também devem ser considerados como a capacidade de identificar *clusters* de tamanhos e densidades diferentes, manipular grande volume de dados, executar o processo

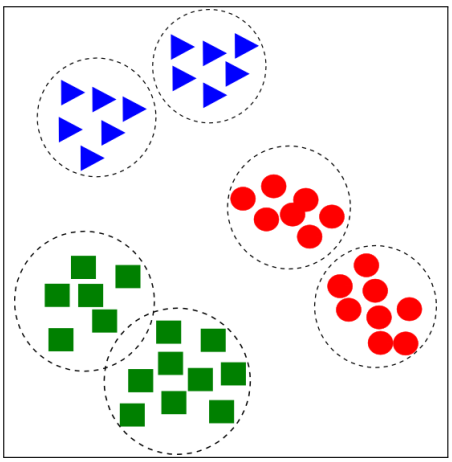


**Figura 2.2:** Exemplo de um conjunto de dados bidimensional separado em diferentes níveis de refinamento. Possíveis interpretações sugerem a existência de três *clusters* formados por objetos de mesma forma geométrica ou a existência de seis *clusters* delimitados por círculos pontilhados.

quando a memória é limitada, etc. A seleção de atributos é comumente aplicada se os *clusters* aparecem apenas quando um subconjunto de atributos é considerado.

### 2.3.4 Algoritmos de agrupamento de dados

Os algoritmos de agrupamento de dados têm por objetivo gerar uma partição do conjunto de dados em um número finito de *clusters*. Eles são categorizados em *hard* ou *fuzzy*. Algoritmos do tipo *hard*, também denotado por *crisp clustering*, geram partições em que cada dado deve pertencer a um único *cluster*. O produto final de um agrupamento do tipo *hard* pode ser representado pela matriz $U$ de ordem $k$ x $n$

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k1} & u_{k2} & \cdots & u_{kn} \end{bmatrix} \tag{2.8}$$

em que $k$ é o número de *clusters* e $n$ é o total de dados. Cada elemento $u_{ij}$ satisfaz as seguintes propriedades (Gan et al., 2007):

$$u_{ij} \in \{0, 1\}, \qquad 1 \leq i \leq k,\ 1 \leq j \leq n \tag{2.9}$$

$$\sum_{i=1}^{k} u_{ij} = 1, \qquad 1 \leq j \leq n, \tag{2.10}$$

$$\sum_{j=1}^{n} u_{ij} > 0, \qquad 1 \leq i \leq k, \tag{2.11}$$

De acordo com a equação 2.9, $u_{ij}$ é uma variável binária e possui valor igual a 1 se o dado $x_j$ pertence ao *cluster* $C_i$. Juntamente com a restrição 2.10 é garantido que cada dado pertença a um único *cluster*. A restrição 2.11 garante que cada *cluster* tenha pelo menos um dado.

Já nos algoritmos do tipo *fuzzy* (não exclusivo), também denotado por *soft clustering*, os dados podem ser associados a mais de um *cluster* com uma certa probabilidade que varia no intervalo de 0 a 1 conforme as equações 2.12, 2.13 e 2.14. Para cada dado $i$, o vetor $u_i$ representa o nível de pertinência dele a cada um dos *clusters*. Os níveis de pertinência expressam a ambiguidade ou o grau de confiança de um dado pertencer a um *cluster* (Oliveira e Pedrycz, 2007). Uma maneira de converter em tipo *hard* é considerar somente o maior nível de pertinência de cada dado.

$$u_{ij} \in [0, 1], \qquad 1 \leq i \leq k,\ 1 \leq j \leq n, \tag{2.12}$$

$$\sum_{i=1}^{k} u_{ij} = 1, \qquad 1 \leq j \leq n, \tag{2.13}$$

$$\sum_{j=1}^{n} u_{ij} > 0, \qquad 1 \leq i \leq k, \tag{2.14}$$

É comum o uso de técnicas de redução de dimensionalidade e seleção de atributos em bases de dados de alta dimensão devido à maldição da dimensionalidade. Contudo, a perda de informação pode levar o algoritmo de agrupamento a detectar *clusters* que não refletem a estrutura original da base de dados. Em geral, algoritmos de agrupamento convencionais têm dificuldades em manipular bases de dados multidimensionais que contêm *clusters* localizados em diferentes subespaços (Gan et al., 2007). Segundo Xu e Wunsch (2005), o algoritmo de agrupamento ideal deve apresentar as seguintes propriedades:

1. detectar *clusters* de formas arbitrárias;
2. baixa complexidade de tempo e espaço para grande volume de dados e bases de alta dimensão;
3. manipula ruídos e *outliers*;
4. pouco influenciado por parâmetros especificados por usuários;
5. a ordem de apresentação dos dados não deve interferir no resultado final;
6. não realiza o processo de aprendizado desde o início para tratar de dados ainda não observados;
7. estima automaticamente o número de *clusters*;
8. aceita mais de um tipo de dado.

Existe uma variedade de algoritmos de agrupamento que diferem entre si na inclusão de uma função específica para avaliação e na estratégia de agrupamento. Alguns deles se enquadram em:

**Algoritmos hierárquicos**

São subdivididos em aglomerativos e divisivos. Na forma aglomerativa, inicialmente cada dado é atribuído a um *cluster* distinto. Em cada passo, os *clusters* mais similares entre si são juntados formando um único *cluster*. O processo finaliza quando todos os dados pertencerem ao mesmo *cluster* formando um dendrograma ou quando um determinado critério de parada é estabelecido. O dendrograma tem uma estrutura similar a uma árvore e exibe as diferentes partições encontradas conforme os níveis de similaridade. De modo oposto, o esquema divisivo começa com todos os dados em um único *cluster* o qual é dividido em partes menores iterativamente até atingir o critério de parada.

Tipicamente, além de medidas de proximidade entre dados apresentadas na seção 2.1.2, os algoritmos aglomerativos ou divisivos usam as medidas de proximidade entre *clusters* tais como:

- *Single linkage*: a distância entre dois *clusters* é a menor das distâncias entre todos os pares de dados de *clusters* distintos. Tende a produzir *clusters* alongados (Jain et al., 1999).
- *Complete linkage*: a distância entre dois *clusters* é a maior das distâncias entre todos os pares de dados de *clusters* distintos. Tende a produzir *clusters* mais compactos.
- *Average linkage*: a proximidade entre dois *clusters* é calculada pela média da distância entre os dados dos dois *clusters*.

Possuem vantagens tais como a facilidade de uso das medidas de proximidade, a capacidade de manipular dados com qualquer tipo de atributo e a não obrigatoriedade do número de *clusters* como parâmetro de entrada (Faceli et al., 2011). Contudo, a estratégia gulosa na criação de *clusters* é uma desvantagem, pois não ocorre otimização após um *cluster* ser formado. Exemplos de algoritmos hierárquicos são: o algoritmo baseado em rede *CHAMELEON* (Karypis et al., 1999) e o algoritmo *BIRCH* (*Balanced Iterative Reducing and Clustering using Hierarchies*) (Zhang et al., 1997) que foi desenvolvido para grandes bases de dados numéricos.

**Algoritmos particionais**

São aqueles que dividem o conjunto de dados em *clusters* não sobrepostos. O particionamento pode ser feito, por exemplo, através do uso de pontos representativos (medóides ou centróides) para representação dos *clusters*. Um medóide corresponde ao dado mais representativo de um *cluster* ao passo que um centróide corresponde ao ponto central de um *cluster*. O algoritmo particional *K-Médias* (Macqueen, 1967) é baseado em centróides, o qual é adequado para encontrar *clusters* esféricos e compactos. O processo de agrupamento do *K-Médias*,

descrito no Algoritmo 1, pode ser tratado como um problema de otimização, no qual a meta é minimizar a distância de cada dado ao centróide mais próximo. Para um conjunto de dados $D = \{x_1, x_2, \ldots, x_n\}$ com $n$ instâncias e $C = \{c_1, c_2, \ldots, c_k\}$ o conjunto de $k$ centróides especificados de antemão, o objetivo é minimizar o valor de $Q$ da equação 2.15. Nesta equação $dist(.,.)$ representa a função de distância e $t_{ih}$ representa uma variável binária com restrições dadas pelas equações 2.16 e 2.17. O valor de $t_{ih}$ é 1 se o dado $x_i$ pertence ao centróide $c_h$, caso contrário o valor é 0. Cada centróide representa o centro de um *cluster* obtido pela média de todos os dados que pertencem a esse centróide. Apesar da ordem de complexidade de tempo ser linear, não é garantido que a solução seja ótima uma vez que o posicionamento inicial dos centróides influencia no resultado final. Os algoritmos baseados em medóides escolhem, no início do processo, um dado para representar cada *cluster* formado por uma regra similar àquela do *K-Médias*. O processo é repetido após a troca de um dado selecionado como medóide por outro ainda não selecionado de tal forma que o agrupamento resultante seja melhor que o anterior. Como um dado é comparado com toda a base de dados para encontrar os pontos representativos, esse esquema não é adequado para grandes bases de dados.

**Algoritmo 1** Algoritmo K-Médias.

Selecionar $k$ dados e posicionar centróides nessas coordenadas.
**repita**
  Gerar $k$ *clusters* atribuindo cada dado ao seu centróide mais próximo.
  Atualizar o posicionamento dos centróides.
**até** as posições dos centróides não mudarem significativamente.

$$Q = \sum_{i=1}^{n} \sum_{h=1}^{k} t_{ih} dist(x_i, c_h) \tag{2.15}$$

$$t_{ih} \in \{0, 1\} \qquad \text{para} \quad i = 1, 2, \ldots, n \quad \text{e} \quad h = 1, 2, \ldots, k \tag{2.16}$$

$$\sum_{h=1}^{k} t_{ih} = 1 \qquad \text{para} \quad i = 1, 2, \ldots, n \tag{2.17}$$

### Algoritmos baseados em densidade

Assumem que *clusters* são regiões densas de dados separados de outros *clusters* por regiões de baixa densidade. Eles analisam a conectividade local dos dados por meio de funções de densidade capazes de detectar *clusters* com formas arbitrárias e, em geral, necessitam de apenas uma única leitura dos dados. *DBSCAN* (*Density-Based Spatial Clustering of Applications with Noise*) (Ester et al., 1996) é um algoritmo que não necessita conhecimento prévio do número de *clusters* e possui dois parâmetros, $\epsilon$ e $MinPts$. Para um dado $x_i$, sua $\epsilon$-vizinhança é definida por $N_{\epsilon-vizinhanca}(x_i) = \{x_j \in D | d(x_i, x_j) \leq \epsilon\}$ que corresponde ao conjunto formado pelos dados que estão na região interna da esfera de raio $\epsilon$ centrada nele. O dado $x_i$ que contém

pelo menos $MinPts$ dados dentro de sua $\epsilon$-vizinhança é rotulado como ponto de núcleo, isto é, $N_{\epsilon-vizinhanca}(x_i) \geq MinPts$. Aqueles que possuem menos de $MinPts$ dados dentro de sua $\epsilon$-vizinhança são rotulados como ponto de borda ou como ponto de ruído. A diferença é que os pontos de borda estão dentro da $\epsilon$-vizinhança de um ponto de núcleo. Como desvantagem, pode-se destacar a dificuldade em agrupar dados de bases de alta dimensão.

## 2.4 Aprendizado semissupervisionado

Aprendizado semissupervisionado (SSL) é um paradigma híbrido que possui características vinculadas tanto com o aprendizado supervisionado quanto com o aprendizado não supervisionado. Surgiu como alternativa aos dois paradigmas convencionais na tentativa de construir melhores modelos mediante o uso de alguns rótulos previamente conhecidos e da imensa porção de exemplos não rotulados. Isso se baseia no fato de que rotular manualmente grande quantidade de exemplos requer considerável esforço humano e que exemplos não rotulados são fáceis de coletar. A coleção de exemplos não rotulados forma estruturas que pode auxiliar no processo de aprendizado.

Resultados empíricos têm mostrado que a combinação de exemplos rotulados e não rotulados pode melhorar o modelo alcançando um desempenho similar ao de muitos modelos supervisionados com a vantagem de utilizar poucos exemplos rotulados. Entretanto, os modelos que seguem este paradigma dependem de uma boa escolha das suposições feitas sobre o conjunto de dados para produzirem resultados satisfatórios. Essas suposições são difíceis de determinar, pois geralmente a distribuição dos dados é desconhecida.

Dentre os principais modos de SSL podem-se listar a classificação semissupervisionada, o agrupamento com restrições, a regressão com dados rotulados e não rotulados, a redução de dimensionalidade com auxílio de dados rotulados, entre outros (Zhu e Goldberg, 2009). Por exemplo, o objetivo da classificação semissupervisionada é prever os rótulos tanto de dados ainda não vistos quanto dos dados não rotulados usados na fase de treinamento.

### 2.4.1 Pressupostos do aprendizado semissupervisionado

Como mencionado anteriormente, os dados não rotulados podem fornecer informações valiosas para o processo de aprendizado. Para isso, suposições devem ser feitas sobre a distribuição desses dados a fim de estimar a fronteira de decisão (Zhu e Goldberg, 2009). Alguns pressupostos adotados pelos métodos de SSL são:

**Pressuposto de suavidade ou *smoothness assumption*:** esta regra analisa o caminho entre dois pontos. Caso o caminho seja de alta densidade, ou seja, os dois pontos estão no mesmo grupo, então seus rótulos são provavelmente equivalentes;

**Pressuposto de agrupamento ou *cluster assumption*:** assume que dois pontos são provavelmente da mesma classe se estiverem no mesmo *cluster*. É possível separar *clusters* em várias classes através de um corte em uma região de baixa densidade. Um exemplo é

reconhecimento dos dígitos manuscritos 0 e 1, descrito em (Chapelle et al., 2006). Esses dígitos estão em classes separadas e a probabilidade de um número estar na fronteira que divide as duas classes é pequena. Logo, a fronteira apresenta baixa densidade;

**Pressuposto de geração de coleções ou *manifold assumption*:** uma estrutura *manifold* é constituída por dados que formam caminhos em regiões de alta dimensionalidade (Breitenbach e Grudic, 2005). Nestes espaços de alta dimensionalidade, a distância geodésia entre dois dados geralmente não é uma reta. Segundo esse pressuposto, tais dados podem ser mapeados para regiões de menor dimensão evitando a maldição da dimensionalidade (Chapelle et al., 2006). Um exemplo de *manifold* é o mapa geográfico que representa o globo terrestre (região de maior dimensão) em um plano (região de menor dimensão). Logo, *manifold* pode ser visto como uma aproximação de uma região de alta dimensionalidade.

Métodos de SSL nem sempre atingem bons resultados na classificação. Por isso, a inclusão de dados não rotulados no processo de aprendizado somente é válida se uma relação for estabelecida entre a distribuição de dados não rotulados e a saída desejada do modelo.

### 2.4.2 Indutivo vs Transdutivo

Quando uma função é gerada a partir de dados apresentados na fase de treinamento, o objetivo do aprendizado é prever rótulos de dados ainda não vistos os quais não fizeram parte do treinamento. A função estimada fará a previsão dos rótulos desconhecidos de dados que fazem parte da fase de teste e que não estavam disponíveis na fase de treinamento. Logo, a função é definida sobre todo o espaço de dados (Chapelle et al., 2006). Esse tipo aprendizado é chamado de indutivo e segue o mesmo esquema do aprendizado supervisionado.

Em casos nos quais todo o conjunto formado por dados rotulados e não rotulados é conhecido, a inferência de uma função sobre todo o espaço de dados é desnecessária (Chapelle et al., 2006). Esse esquema é denominado aprendizado transdutivo no qual uma função de menor complexidade pode ser estimada e a previsão dos rótulos ocorre na fase de treinamento. Mais adiante serão descritos os métodos SSL baseados em redes que são exemplos deste esquema.

### 2.4.3 Modelos generativos

Modelos de mistura são algoritmos de SSL que visam decompor o conjunto de dados em classes baseando-se nas distribuições formadas pelos dados não rotulados. Dado que as distribuições sejam conhecidas, o aprendizado ocorre com a determinação dos parâmetros desconhecidos dessas distribuições (Zhu e Goldberg, 2009). Formalmente, busca-se maximizar a probabilidade condicional $P(y|x)$, isto é, atribuir ao dado $x$ o rótulo da classe $y$ mais provável. Por definição, tem-se que $p(y|x) \in [0, 1]$ e $\sum_{y'} p(y'|x) = 1$.

Os modelos generativos utilizam a regra de Bayes (fórmula 2.18) na qual o somatório é feito sobre todas as classes $y'$ do conjunto de dados. O valor $p(y)$ corresponde a probabilidade da

classe $y$, isto é, a proporção de dados que pertencem à classe $y$. Dado um conjunto de modelos $\{p(x|y,\theta)\}$, deve-se determinar aquele cujo conjunto de parâmetros $\theta$ da distribuição apresente o menor erro de classificação. Quando a distribuição dos dados é gaussiana, uma das maneiras de estimar o conjunto de parâmetros $\theta$ é por meio do algoritmo de Maximização de Expectativa (*Expectation Maximization*).

$$p(y|x) = \frac{p(x|y)p(y)}{\sum_{y'} p(x|y')p(y')} \tag{2.18}$$

### 2.4.4 Separação de baixa densidade

O pressuposto de algoritmos de separação de baixa densidade é que as classes são separadas por regiões de baixa densidade tal que a probabilidade de existir algum dado posicionado próximo à fronteira de decisão é pequena. O método *TSVM* (*transductive support vector machines*) (Vapnik, 1998) é um exemplo clássico dessa categoria. Trata-se de uma abordagem transdutiva do algoritmo SVM, pois não necessita aprender uma regra geral de aprendizado. As margens são maximizadas com auxílio não somente dos dados rotulados como também dos dados não rotulados que revelam informações importantes sobre os *clusters*. Tais informações orientam as margens para longe das regiões densas de maneira que a fronteira entre classes esteja de acordo com a fronteira entre *clusters*. Logo, o pressuposto de suavidade deve ser cumprido. Pesquisas têm destacado o algoritmo *TSVM* como uma solução promissora no contexto de classificação de textos em tópicos. Entretanto, ainda é um desafio encontrar uma solução ótima global em tempo razoável para o problema de otimização das margens que envolvem muitos dados. Consequentemente, algoritmos que geram soluções aproximadas têm sido propostos.

### 2.4.5 Métodos baseados em redes

Como o próprio nome sugere, algoritmos baseados em redes representam bases de dados na forma de redes. Usualmente, essa representação mapeia dados como vértices e a similaridade entre dois vértices com a presença de uma aresta que os conecta. Quando as arestas são ponderadas, os pesos indicam o nível de similaridade ou a distância entre os dados. Vários problemas originalmente representados em forma de redes evidenciam o interesse nesses métodos como: otimização da infraestrutura de rede de comunicações, classificação de genes e interação entre proteínas, identificação de grupos em redes sociais que compartilham o mesmo interesse, estudo da propagação de epidemias em redes de interações entre indivíduos, entre outros (Schaeffer, 2007).

Rede é um conjunto $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ construído a partir de uma base de dados, em que $\mathcal{V} = \{v_1, \ldots, v_n\}$ representa o conjunto finito não vazio de vértices e $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}, i \neq j\}$ representa o conjunto de arestas, cada uma conectando um par $(v_i, v_j)$ não ordenado de vértices. Cada vértice $v_i$ está associado a um objeto da base de dados e cada aresta $e = (v_i, v_j)$ possui um peso $W_{ij}$ que representa numericamente a similaridade ou dissimilaridade entre o par de

vértices. Em uma matriz de similaridade, maior será a similaridade entre os vértices $v_i$ e $v_j$ quanto maior o valor de $W_{ij}$.

Existem diversas formas para gerar a matriz de pesos $W$. Uma delas é por meio do *kernel gaussiano* (expressão 2.19) que gera uma rede totalmente conectada e ponderada. O parâmetro $\sigma$, definido pelo usuário, indica a rapidez do decréscimo do peso e, por convenção, $W_{ii} = 0$. Outra forma é conectar cada dado com os seus $k$ vizinhos mais próximos gerando uma rede $k$NN. Essa abordagem pode gerar mais que $k$ arestas em um vértice e não garante que a rede seja conexa. Também existe a rede denominada $\epsilon$NN obtida com a inclusão de arestas entre vértices que estão a uma distância menor que um valor $\epsilon$. Essas duas últimas abordagens permitem que a geração de uma rede não ponderada de forma que a matriz de pesos $W$ possua apenas valores binários: $0$ (aresta ausente) e $1$ (aresta presente).

$$W_{ij} = exp(-\frac{||x_i - x_j||^2}{2\sigma^2}) \tag{2.19}$$

Métodos dessa categoria podem ser desenvolvidos com a criação de uma função constituída por dois termos: uma função de perda e um termo regularizador. A função de perda (*loss function*) é responsável por estabelecer a restrição de proximidade entre os dados tal que quaisquer dois dados provavelmente terão o mesmo rótulo se estiverem próximos entre si. O termo regularizador por sua vez garante que o pressuposto de suavidade na rede seja cumprido. A seguir serão brevemente descritos alguns métodos.

#### Corte mínimo

Corte mínimo tem por objetivo particionar a rede em subredes tal que elementos com rótulos distintos não fiquem na mesma subrede. No caso de uma classificação binária com uma classe positiva (fonte) e outra negativa (sumidouro), o objetivo é bloquear o fluxo de fontes para sumidouros por meio da eliminação (corte) de arestas cuja soma dos pesos seja mínima (Zhu, 2005). Então, exemplos conectados com os vértices fontes são classificados como positivos e os exemplos conectados com vértices sumidouros são classificados como negativos.

Considerando que os $l$ primeiros objetos do conjunto de dados estejam previamente rotulados, o critério de corte é definido pela regra de minimização da equação 2.20 na qual a função $f(x)$ retorna $-1$ (classe negativa) ou $1$ (classe positiva) para um dado $x$ qualquer. O primeiro termo dessa regra corresponde à função de perda que fixa o valor da função $f(x)$ para os objetos com rótulos conhecidos e assume que $\infty \cdot 0 = 0$. O último termo da regra (regularizador) envolve apenas os objetos com rótulos desconhecidos e contribui com o peso das arestas não removidas cujos vértices não pertencem a mesma classe.

$$min_{f:f(x)\in\{-1,1\}} \infty \sum_{i=1}^{l}(y - f(x_i))^2 + \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 \tag{2.20}$$

### Propagação de rótulos

A ideia da propagação de rótulos é classificar todos os elementos da rede empalhando o rótulo dos vértices, cuja classe é conhecida, aos vértices vizinhos. Esse processo é repetido até que todos os vértices estejam rotulados.

**Algoritmo 2** Propagação de rótulos

Calcular a matriz de peso $W$ com diagonal zero.
Calcular a matriz diagonal $D$ tal que $D_{ii} = \sum_j W_{ij}$
Escolher um parâmetro $\alpha \in (0, 1)$ e um $\epsilon > 0$
$\mu = \frac{\alpha}{1-\alpha} \in (0, +\infty)$
Calcular a matriz diagonal $A$ tal que $A_{ii} = I_l(i) + \mu D_{ii} + \mu\epsilon$
Inicializar $\hat{Y}^{(0)} = (y_1, \ldots, y_l, 0, 0, \ldots, 0)$
Iterar $\hat{Y}^{(t+1)} = A^{-1}(\mu W \hat{Y}^{(t)} + \hat{Y}^{(0)})$ até convergir para $\hat{Y}^{(\infty)}$
Rotular o ponto $x_i$ de acordo com o sinal de $\hat{y_i}^{(\infty)}$

O algoritmo 2 descreve um esquema de propagação de rótulos proposto em (Chapelle et al., 2006). Neste algoritmo, a etapa de iteração é realizada pela equação 2.21 para os dados rotulados ($i \leq l$) e pela equação 2.22 para os dados não rotulados ($l + 1 \leq i \leq n$). O somatório contido no numerador de ambas as frações corresponde a média ponderada dos rótulos dos vizinhos do objeto $x_i$ e o termo $\epsilon$ evita que o denominador das frações seja nulo. Nota-se que na primeira equação aparece um fator $\frac{1}{\mu}$ que serve para atribuir um grau de confiabilidade para o rótulo do objeto $x_i$. Quando o valor de $\mu$ é pequeno, o rótulo desse objeto dificilmente será mudado. Logo, o algoritmo permite que um exemplo previamente rotulado possa troca seu rótulo.

$$\hat{y_i}^{(t+1)} = \frac{\sum_j W_{ij} \hat{y_i}^{(t)} + \frac{1}{\mu} y_i}{\sum_j W_{ij} + \frac{1}{\mu} + \epsilon} \tag{2.21}$$

$$\hat{y_i}^{(t+1)} = \frac{\sum_j W_{ij} \hat{y_i}^{(t)}}{\sum_j W_{ij} + \epsilon} \tag{2.22}$$

Szummer e Jaakkola (2002) propuseram um algoritmo de propagação de rótulos fundamentado na caminhada aleatória de Markov. A caminhada definirá a probabilidade de cada vértice pertencer às classes existentes. Define-se a expressão $\mathbb{P}^t(y = 1|i)$ como a probabilidade do vértice $i$ pertencer à classe 1 dado que um andarilho inicie sua caminhada a partir de um vértice dessa classe e termine no vértice $i$ após $t$ passos. Como esse algoritmo depende de um parâmetro $t$, outra opção é iniciar a caminhada a partir de um vértice não rotulado e calcular o número de passos necessários para chegar até um vértice rotulado. A probabilidade de transição do andarilho visitar um vértice $j$ partindo de um vértice $i$ é dada pela expressão 2.23 na qual $W$ corresponde a matriz de similaridade e o denominador é calculado pelo somatório da similaridade entre o vértice $i$ e seus adjacentes. Assim, em um problema de classificação envolvendo

apenas duas classes, o vértice $i$ receberá, após um número suficiente de iterações, o rótulo da classe cuja probabilidade de visita do andarilho é superior a $0.5$.

$$\mathbb{P}_{ij} = \frac{W_{ij}}{\sum_{k} W_{ik}} \tag{2.23}$$

**Modelo de competição e cooperação de partículas**

O modelo de competição e cooperação de partículas é um método que realiza a propagação de rótulos por meio partículas que caminham sobre a rede dominando o maior número possível de vértices defendendo seus territórios da invasão de outras partículas. A descrição do método dada a seguir é semelhante as abordagens propostas pelos autores Silva e Zhao (2012) e Breve et al. (2011).

Cada classe do conjunto de dados é representa por um time de partículas que cooperam entre si propagando rótulos de instâncias da mesma classe. A competição ocorre da disputa entre partículas de times diferentes por vértices da rede. Assim, cada partícula é encarregada de disseminar o rótulo do vértice no qual ela foi posicionada no início do processo. Esses vértices previamente rotulados são chamados de *vértices casa*. A classificação dos vértices cujos rótulos são desconhecidos pode ser feita analisando o número de visitas que cada vértice recebeu. Então, uma possível estratégia é atribuir a cada vértice o rótulo da classe do time de partículas que mais o visitou. Um detalhe importante é que a rede construída seja conexa ou exista pelo menos um vértice previamente rotulado em cada uma das subredes conexas para fins de propagação de rótulo.

Existem dois tipos de movimento de uma partícula: movimento aleatório e movimento preferencial. O tipo de movimento realizado, em um determinado instante, depende de um parâmetro $p_{grd}$ com $0 \leq p_{grd} \leq 1$. Uma partícula tem probabilidade $p_{grd}$ de realizar um movimento preferencial e probabilidade $1 - p_{grd}$ de realizar um movimento aleatório. O movimento aleatório permite que a partícula visite um vértice vizinho $v_j$ a partir de um vértice $v_i$ com probabilidade constante, desde que aquele vértice adjacente não esteja ocupado por uma outra partícula. A probabilidade invariante no tempo de visitar um vértice adjacente é definida pela seguinte expressão,

$$\mathbb{P}^{(k)}_{rand}(i,j) \triangleq \frac{a_{ij}}{\sum_{o} a_{io}}, \tag{2.24}$$

em que $a_{ij}$ é 1 se, e somente se, o vértice $j$ é um vizinho do vértice $i$, isto é, $j \in NNk(i)$; caso contrário ele é zero. Dada uma matriz de similaridade $W$, o movimento preferencial permite que a partícula visite um vértice dominado pelo seu próprio time. Neste caso, a probabilidade invariante no tempo é

$$\mathbb{P}^{(k)}_{pref}(i,j) \triangleq \frac{W_{ij}}{\sum\limits_{o \in NNk(i)} W_{io}}, \tag{2.25}$$

em que $i$ e $j$ são os índices dos vértices de saída e entrada e pertencem ao mesmo time. Em (Breve et al., 2011), essa probabilidade depende também de outros dois fatores: nível de energia do vértice e distância da partícula em relação ao seu *vértice casa*. O nível de energia de um vértice é um vetor de dimensão igual ao número de times, cujos valores somam 1. Esse vetor indica a força de dominância de cada time sobre aquele vértice. Toda vez que uma partícula de um time $m$ visita o vértice, o valor da dimensão $m$ é incrementado e os demais são decrementados a fim de manter os valores normalizados. Além do nível de energia, os autores consideram que cada partícula atua apenas em uma região limitada pela distância em relação ao *vértice casa*. Na implementação dos autores Silva e Zhao (2012), a probabilidade preferencial é diretamente proporcional ao número relativo de visitas.

A caminhada das partículas é repetida por um número suficientemente grande de etapas tal que todo vértice seja dominado por um time de partículas. Seja $\mathcal{C}$ o conjunto de classes e $N$ um vetor $c$-dimensional que descreve o número de visitas que o vértice $v_i$ recebeu por todas as partículas do time m até a iteração $t$. Quando um vértice é visitado por uma partícula que pertence ao time m, seu número de visitas $N$ é aumentado em uma unidade. Na iteração $t$, o vértice $v_i$ é dominado pelo time de partículas de índice m se, e somente se, $m = \underset{c \in \mathcal{C}}{argmax}(N_i^{(c)}(t))$. Quando o processo de competição e cooperação termina, cada vértice não rotulado recebe o rótulo da classe do time de partículas cujo número de visitas é máximo.

Cada partícula possui um nível de energia $E$ o qual é atualizado em cada iteração, com $0 < E < 1$. Se uma partícula tenta visitar um vértice dominado por outro time, seu nível de energia é reduzido conforme a primeira linha da expressão 2.26, em que *owner* é uma função booleana que retorna 1 se o vértice $i$ é dominado pelo time de partículas de índice $m$, ou 0 caso contrário. Isso significa que uma partícula do time $m$ não pode ocupar o vértice $i$ enquanto o valor do elemento $N$ não for o máximo do vetor $N$. Se o nível de energia reduzir a zero, ele recebe o valor $0.5$ e a partícula é automaticamente colocada sobre o seu *vértice casa*. Em contrapartida, se o vértice já está dominado pelo mesmo time da partícula, o nível de energia é incrementado conforme a segunda linha da expressão 2.26. Em resumo, a primeira regra de atualização evita que partículas invadam território inimigo diminuindo o nível de energia enquanto a segunda regra ajuda as partículas do mesmo time a proteger seus territórios de outros times. Os autores definem um critério de parada, bem como os valores dos parâmetros $p_{grd}$ e $\Delta$.

$$E^{(p)}(t) = \begin{cases} max(0, E^{(p)}(t-1) - \Delta), \text{ if } \neg owner(i, m, t) \\ \\ min(1, E^{(p)}(t-1) + \Delta), \text{ if } owner(i, m, t) \end{cases} \tag{2.26}$$

Seja $N_i(t) = \{N_i^1(t), N_i^2(t), \ldots, N_i^c(t)\}$ um vetor c-dimensional em que $N_i^m(t)$ descreve o número de visitas que um vértice $v_i$ recebeu por todas as partículas do time de índice $m \in \{1, 2, \ldots, c\}$ até a iteração $t$. Quando um vértice $v_i$ é visitado por uma partícula $p$ do time $m$, seu número de visitas $N_i^{(m)}$ é incrementado em uma unidade. Na iteração $t$, o vértice $v_i$ é dominado pelo time de índice $m$ se, e somente se, $m = \underset{c \in \mathcal{C}}{argmax}(N_i^{(c)}(t))$. Quando o processo termina,

cada objeto não rotulado recebe o rótulo do time de partículas cujo número de visitas é máximo, ou seja, o rótulo de um vértice $v_i$ não rotulado é dado por $y_i = \underset{m \in \mathcal{C}}{argmax}(N_i^{(m)}(t))$. Esse processo é similar ao modelo proposto por Breve et al. (2011), cujo processo de classificação é guiado pelo vetor de energia.

## 2.5 Considerações Finais

Neste capítulo, foram descritos os principais conceitos sobre aprendizado de máquina destacando três categorias de aprendizado de máquina: aprendizado supervisionado, não supervisionado e semissupervisionado. A primeira delas inclui métodos que fazem uso de dados rotulados para construção de um modelo que seja capaz de prever os rotulados de objetos ainda não observados. A segunda categoria diz respeito aos métodos que não fazem uso de dados rotulados e visam identificar como os dados estão organizados. Tal processo é feito, por exemplo, por meio de métodos de agrupamento de dados. Foram ainda discutidos os tipos de *clusters*, as etapas do agrupamento de dados, a etapa de validação de resultados e alguns tipos de algoritmos de agrupamento de dados. Em geral, o desempenho é inferior se comparado aos métodos supervisionados, mas há grande aplicabilidade na indústria.

Na última seção, discutiu-se sobre uma categoria híbrida de aprendizado de máquina: o aprendizado semissupervisionado. Foi visto que nem sempre a utilização de dados não rotulados pode melhorar o desempenho dos modelos e que a eficiência depende do cumprimento de pressupostos em relação à distribuição dos dados. Quando satisfeita tais exigências, métodos desse paradigma têm obtido resultados comparáveis as abordagens supervisionadas com a vantagem de necessitar de poucos dados rotulados para a construção do modelo. Dentre os principais métodos de aprendizado semissupervisionado, destacam-se aqueles baseadas em redes. Em particular, descreveu-se em maiores detalhes o funcionamento geral do modelo de competição e cooperação de partículas contido nos artigos (Breve et al., 2011; Silva e Zhao, 2012). Segundo os autores, o modelo tem a capacidade de identificar *clusters* de formas arbitrárias e baixo custo computacional em determinadas situações.

Capítulo
3

# Detecção de Outliers

Segundo D. M. Hawkins (1980) *outlier* é uma observação que desvia-se muito das demais observações sob suspeita de ter sido gerada por um mecanismo diferente. Esse assunto tem sido extensivamente investigado em diversas áreas da computação como em segurança da informação, análise de desempenho de redes de computadores, mineração de dados web, etc.

Geralmente, *outliers* estão associados a eventos raros e rótulos desses dados são escassos. No contexto de aprendizado de máquina, a disponibilidade de rótulos influencia na construção de um classificador. Na abordagem supervisionada, cujo objetivo é gerar um classificador capaz de discernir dados normais de *outliers*, é necessário um número suficiente de exemplos rotulados de ambas as classes. É comum associar *outliers* a classe minoritária (rara) do conjunto de dados com atribuição de um custo maior no erro de classificação desses dados. Isso decorre da não garantia de boa precisão do classificador mesmo que ele obtenha $99\%$ de acerto uma vez que os *outliers* podem estar incluídos no $1\%$ de dados classificados incorretamente. Desvantagens da abordagem supervisionada incluem a dificuldade na construção de um classificador usando amostras desbalanceadas e a necessidade de manter o classificador atualizado quando *outliers* sofrem mudanças de comportamento (Su e Tsai, 2011). Por outro lado, técnicas de aprendizado não-supervisionado não necessitam de conhecimento dos rótulos e fazem duas suposições. A primeira delas é que dados normais estão em maior número e a segunda suposição é que *outliers* sejam suficientemente diferentes dos dados normais. Como visam encontrar padrões nos dados agrupando aqueles que apresentam características semelhantes, requerem tais suposições. Por último, a abordagem semi-supervisionada busca obter melhores resultados em relação ao aprendizado não supervisionado fazendo uso de rótulos da classe normal para identificação de *outliers*. Comumente, apenas dados normais possuem rótulos disponíveis e por isso abordagem semissupervisionada têm maior valor prático em relação a supervisionada.

## 3.1 Conceitos gerais

Não há uma definição única para o termo *outlier*. Segundo Barnett e Lewis (1995), tal termo refere-se a uma observação inconsistente com o restante dos dados. Theodoridis e Koutroumbas (2008) o descrevem como um ponto que está distante da média de uma variável aleatória. Grubbs (1969) o definiu como uma observação que parece desviar-se acentuadamente se comparada aos demais membros da amostra na qual ela ocorre. Todas as definições anteriores são genéricas e uma explicação mais minuciosa depende da aplicação e da técnica de detecção empregada.

Os fatores mais comuns associados ao surgimento de *outliers* são: erros de digitação, defeitos em instrumentos de medição, variação natural na distribuição dos dados, comportamento fraudulento, falhas em sistemas e mudanças de comportamento de sistemas (Hodge e Austin, 2004). Na estatística, um erro de observação ou de arredondamento pode gerar um *outlier* (Morettin e Bussab, 2003).

*Outlier* é um assunto bastante explorado pela comunidade científica, pois contém informações de grande valor em muitas aplicações reais e favorece a compreensão das características dos conjuntos de dados. Em alguns casos, tais informações precisam ser identificadas o mais rápido possível para reduzir prejuízos como, por exemplo, a perda de produtividade causada por falhas em máquinas de produção. A estratégia deve ser preventiva tal que o comportamento das máquinas seja analisado ao longo do tempo e seja possível detectar condições de funcionamento anormais que antecedem uma falha. Dado que condições anormais diferem significativamente do comportamento normal da máquina, esse tipo de problema está relacionado à detecção de *outliers*.

Uma questão polêmica sobre esse tema é a discordância entre o significado de *outlier* e ruído. Ao contrário de ruídos, os *outliers* contêm informações importantes para descrição do conjunto de dados (Chandola et al., 2009; Tan et al., 2005). Ruídos prejudicam a análise do conjunto de dados e, por isso, removê-los é o processo mais comum. Para Quinlan (1986), ruídos são instâncias classificadas incorretamente ou erros em valores de atributo enquanto *outlier* é um conceito mais amplo que inclui ruídos. Em (Alpaydin, 2010), ruído é descrito como uma anomalia não desejada no dado causada por imprecisão nos atributos de entrada, erro na rotulação dos dados (*teacher noise*) ou até por atributos de dados que foram excluídos, mas afetam a rotulação. Ruído ainda pode ser caracterizado por valores que estão fora do domínio. O tratamento desses dados depende da área de aplicação. Como ruídos parecem ser similares aos *outliers*, frequentemente ambos são tratados de modo indistinto.

Detecção de *outliers* refere-se ao problema de encontrar padrões que exibem um comportamento diferenciado em relação à maioria dos dados (Chandola et al., 2009). Esse problema consiste apenas na etapa de identificação e o tratamento desses dados ocorre em uma etapa posterior, normalmente executado por um especialista. *Noise removal*, *Novelty detection*, *anomaly detection* e *exception mining* são tópicos vinculados ao mesmo problema (Su e Tsai, 2011) e referem-se as observações anormais de um ponto de vista diferente denotando-as como obser-

vações discordantes, dados contaminantes, exceções, ruídos, anomalias ou simplesmente como *outliers*. Abaixo, uma breve descrição de cada um desses tópicos:

***Noise removal*:** remoção de dados que não contém informação útil ou que prejudicam a análise do conjunto de dados. Basta a presença de um *outlier* para distorcer a média e o desvio padrão. Esse tópico geralmente está associado à etapa de pré-processamento.

***Novelty detection*:** detectar padrões ainda não observados. Em métodos de aprendizado de máquina, esses padrões são dados que não fizeram parte do conjunto de treinamento do modelo e diferem significativamente daqueles já conhecidos. Eles podem ser incluídos no conjunto de exemplos normais após serem detectados.

***Anomaly detection*:** o objetivo é detectar dados que desviam do comportamento normal. Embora dados com comportamento anormal sejam raros, eles podem ocorrer com frequência dependendo do tamanho do conjunto de dados.

***Exception mining*:** detectar padrões raros ou eventos raros a partir de um grande volume de dados (Agyemang et al., 2006).

Situações nas quais já se conhece quais dados são *outliers*, isto é, quando as classes dos dados estão disponíveis, os *outliers* representam a classe positiva em uma avaliação de desempenho porque o interesse está na detecção desses dados. Acurácia, precisão, revocação, especificidade e medida-f estão entre as medidas frequentemente utilizadas para esse propósito. Outra possibilidade para análise é construir curvas *ROC*. A meta é obter bom desempenho de precisão e alarme falso. A precisão é definida como a taxa com que dados encontrados são realmente *outliers* e o alarme falso é usado para verificar a taxa de dados normais classificados como *outliers*. O custo do erro de classificação dependerá do tipo de aplicação.

### 3.1.1 Tipos de Outliers

Segundo Chandola et al. (2009), os *outliers* podem ser divididos em três tipos: pontual, contextual e coletivo.

***Outlier* pontual:** objeto considerado anormal em relação aos outros objetos. Um exemplo de *outlier* pontual é o ponto $p$ no canto direito mostrado na Figura 3.1. Objetos normais tendem a estar próximos entre si enquanto um *outlier* pontual está distante de todos os demais pontos.

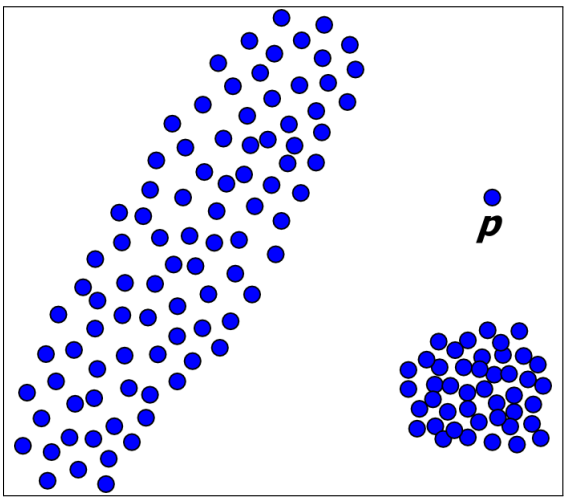



**Figura 3.1:** Representação de um conjunto de dados bidimensional com dois *clusters* de densidades diferentes. O ponto $p$ é um *outlier* pontual.

***Outlier* contextual:** objeto considerado anormal em relação a um específico contexto e não necessariamente a todo o conjunto de dados. Cada objeto é analisado através de atributos contextuais que determinam o contexto do objeto e atributos comportamentais que determinam as demais características como a anormalidade do objeto naquele contexto. Pode-se realizar, por exemplo, agrupamento de dados para definir o contexto. Já em problemas de classificação, *outlier* contextual é visto como um dado significativamente diferente dos demais dados da mesma classe (He et al., 2004). O *outlier* contextual é frequentemente estudado em séries temporais, dados espaciais e redes. Na detecção de fraudes em cartões de crédito é possível analisar o valor mensal gasto por um indivíduo. Gasto excessivo em mês que não possui datas comemorativas pode ser um indício de fraude ao passo que o mesmo valor poderia ser considerado normal em meses como dezembro (natal) e abril (páscoa). Um exemplo de *outlier* contextual em rede foi dado por J. Sun et al. (2005). A Figura 3.2 mostra a representação de uma rede bipartida composta por dois conjuntos distintos de vértices claramente diferenciados pela forma geométrica. Considerando os vértices em forma de círculo como autores e os vértices em forma de quadrado como artigos publicados por autores conectados a eles, o vértice $T$ provavelmente é um *outlier* contextual, pois conecta com vértices (autores $A$ e $B$) de diferentes vizinhanças, as quais são delimitadas pela elipse pontilhada. Essa interpretação é devido à suposição de que autores de comunidades diferentes raramente publicam artigos em conjunto.

***Outlier* coletivo:** corresponde a um subconjunto de objetos que são anormais quando formam grupos. Na série temporal de um eletrocardiograma, a ocorrência seguida de observações com mesmo valor é rara e não está em conformidade com o restante dos dados. Essas observações anormais poderiam indicar contração prematura do átrio. Em redes, *outlier* coletivo é representado por uma subrede, cujo grau de anormalidade pode ser medido pela quantidade de subestruturas comuns que aparecem nela. Subredes com subestruturas mais comuns são, em geral, menos anormais que aquelas com subestruturas incomuns

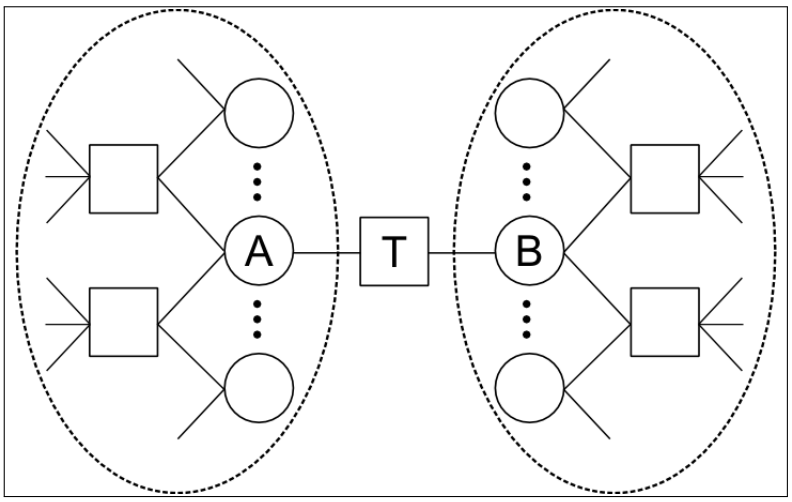



**Figura 3.2:** Representação de uma rede bipartida formada por dois conjuntos distintos de vértices na forma de quadrado e na forma de círculo. Se as elipses pontilhadas delimitam vértices de mesma vizinhança, o vértice $T$ é considerado um *outlier* contextual. Figura adaptada de J. Sun et al. (2005).

(Noble e Cook, 2003). Considerando subredes de tamanho três delimitadas por círculos pontilhados (Figura 3.3), a subestrutura $A \rightarrow B$ é comum pois ocorre em três subredes. Logo, a subrede de três vértices localizada no canto inferior direito é mais anormal que as outras. Técnicas destinadas a detectar esse tipo de dado têm recebido grande atenção nos últimos anos devido à descoberta de sua aplicabilidade.

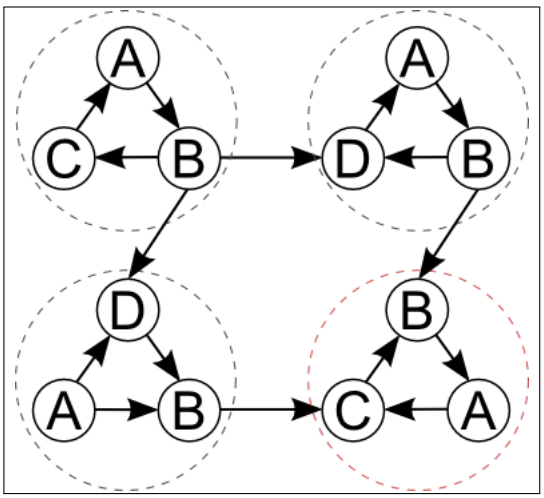



**Figura 3.3:** Representação de um *outlier* coletivo em rede. Das subredes delimitadas por círculos pontilhados de tamanho três, aquela localizada no canto inferior direito é a mais anormal por não conter a subestrutura mais comum $A \rightarrow B$.

### 3.1.2 Classificação de *Outliers*

As técnicas de detecção de *outliers* geram uma das seguintes saídas:

**Rótulo:** cada dado é rotulado como normal ou como *outlier*. Define a quantidade esperada de *outliers* ou um limiar que separa dados *outliers* de dados normais.

**Ranqueamento ou Escore:** atribuição de um grau de anormalidade para cada dado não especificando quais dados são *outliers* para que essa análise seja feita em uma etapa posterior. Uma lista contendo os dados em ordem decrescente do grau de anormalidade é produzida tal que os dados próximos ao topo da lista são os potenciais *outliers*. A saída pode ser convertida para binária com a definição de um limiar para o grau de anormalidade.

### 3.1.3 Aplicações

Os interesses no estudo de detecção de *outliers* estão relacionados à redução de prejuízos e permitir uma melhor análise dos dados. Antigamente *outliers* eram simplesmente removidos do conjunto de dados, pois acreditava-se que eles não continham informações úteis. Entretanto, verificou-se que esses dados favorecem o processo de mineração de dados em aplicações tais como:

**Detecção de fraudes:** o objetivo é identificar crimes que ocorrem em organizações comerciais como fraudes de cartões de crédito, de seguro e de celulares. Duração de chamadas, destino, frequência de uso são exemplos de características usadas para fraudes de celulares. No caso de roubos ou uso incomum de cartões de crédito, pode ser levado em consideração o histórico de transações e localização geográfica. Boa parte das aplicações faz uso de técnicas de detecção que operam em tempo real através do monitoramento de atividades. O alarme deve ser disparado tão logo que um evento anormal tenha ocorrido. Casos investigados por especialistas podem servir como exemplos na construção de modelos supervisionados, os quais são treinados a partir de exemplos de fraudes e transações normais. Já os modelos não supervisionados usam outras formas de detectar fraudes sendo capazes de detectar tipos de fraudes ainda não observados (Kou et al., 2004).

**Processamento de imagens:** detectar regiões anormais em imagens como, por exemplo, detecção de movimento em sistemas monitorados por câmeras de vigilância, manchas em fotografias de raio-x, na análise de fotografias fornecidas por satélites, entre outras (Singh e Markou, 2004).

**Detecção de intrusão:** no contexto de redes de computadores, intrusão refere-se ao ato de comprometer a estabilidade da rede ou a segurança da informação contida em computadores (Portnoy et al., 2001). A tarefa é desenvolver sistemas aptos a detectar intrusões. Tais sistemas podem ser baseados em rede se a detecção de intrusão for feita pelo monitoramento do tráfego com auxílio de dispositivos de rede ou podem ser baseados em *host* se o monitoramento é realizado em uma específica máquina através de softwares que

verificam atividades de processos e uso de arquivos. O modelo construído a partir de dados de treinamento que difere essencialmente na presença ou ausência de dados rotulados. No caso supervisionado, o modelo detecta intrusões com auxílio dos exemplos de intrusão utilizados na etapa de treinamento. Os modelos devem ser treinados novamente se surgirem novos tipos de intrusão. Quando não há rótulos disponíveis, o objetivo é detectar observações que desviam dos dados normais. Assim, supõe-se que a quantidade de instâncias normais é muito maior que a quantidade de instâncias de intrusão e que as intrusões são suficientemente diferentes dos dados normais. Detecção de intrusão é um tópico de constante pesquisa e diversas técnicas podem ser encontradas na literatura (S. Hawkins et al., 2002), (Li et al., 2007), (Portnoy et al., 2001).

**Medicina:** detectar condições anormais em equipamentos médicos e em pacientes. Pacientes que apresentam condições anormais podem estar com problemas de saúde. O desafio é identificar essas condições anormais assim que ocorrerem para diagnosticar o paciente a tempo ou para evitar exames médicos desnecessários.

**Detecção de dados rotulados incorretamente:** Certos modelos são construídos a partir de exemplos rotulados por especialistas. É possível que alguns desses dados sejam rotulados incorretamente. Modelos capazes de prever os verdadeiros rótulos desses dados fazem parte desse tipo de aplicação.

Além das aplicações citadas acima, destacam-se a análise de desempenho de redes, aprovação de empréstimos, identificação de novas estruturas moleculares, detecção de defeitos em maquinaria e o monitoramento de séries temporais em aplicações de segurança (Hodge e Austin, 2004).

## 3.2 Técnicas de Detecção de Outliers

Grande esforço tem sido dispendido na criação de técnicas que sejam pouco influenciadas por parâmetros de entrada. Em determinados casos, obter o valor ótimo de um parâmetro não é uma tarefa trivial, assim como a definição da fronteira entre dados normais e *outliers*. Não existe uma técnica de detecção eficiente para todo tipo de aplicação. As técnicas de detecção de *outliers* podem ser aplicadas em remoção de ruídos, detecção de padrões ou eventos raros, detecção de padrões ainda não observados, detecção de dados que desviam do comportamento normal, etc. A complexidade computacional é outro assunto que deve ser considerado. Com o aumento da informação disponível, a capacidade de manipular grande volume de dados e dados de alta dimensão é uma habilidade desejada.

Diversos fatores são considerados para categorizar as técnicas de detecção de *outliers*: o tipo de *outlier* encontrado, a capacidade de manipular grande volume de dados e conjuntos de alta dimensão, a natureza dos dados como sequenciais e espaciais, a disponibilidade de rótulos, necessidade ou não de parâmetros, etc. Neste trabalho, procurou-se separar as categorias em estatística, baseadas em distância, baseadas em agrupamento de dados, baseadas na teoria da

informação e baseadas em redes. Nessa última categoria as técnicas tratam apenas de conjuntos de dados em forma de redes e utilizam artifícios empregados por outras categorias.

### 3.2.1 Técnicas estatísticas

Os primeiros trabalhos de detecção de *outliers* vieram da estatística. Durante a análise de dados de uma população, foi verificado que certas observações (*outliers*) não seguiam a mesma distribuição da maioria dos dados. A partir desta constatação, duas hipóteses foram definidas: *outliers* são observações que seguem uma distribuição diferente das demais observações ou ocorrem em regiões de baixa probabilidade de uma mesma distribuição. A primeira hipótese pode estar relacionada a dados contaminantes, ou seja, observações de outra distribuição, que podem afetar a média e o desvio padrão do conjunto de dados. Quando supõe que os dados são modelados por uma distribuição conhecida, são usados estimadores estatísticos dos parâmetros da distribuição os quais são obtidos por amostras e cada observação possui um grau probabilístico de anormalidade que pode ser definido, por exemplo, pelo inverso da função densidade de probabilidade (Chandola et al., 2009).

Grande parte das ferramentas estatísticas para detecção de *outliers* é desenvolvida para dados univariados. Apesar dessa limitação, as ferramentas estatísticas apresentam boa precisão. Uma maneira de detectar observações anormais em distribuições gaussianas é através da técnica de Grubbs (1969) (equação 3.1) que calcula a anormalidade de um dado pela diferença entre o valor de um dado $x_i$ e a média $\mu$ do conjunto de dados dividida pelo desvio padrão $\sigma$. Essa equação é chamada de $z\_score$ e mede a distância do valor $x_i$ a média da distribuição. Após um *outlier* ser detectado, ele é removido e o processo é repetido novamente até nenhum dado ser removido. Quanto maior a quantidade de dados mais preciso é o resultado. Vale ressaltar que os *outliers* podem formar pequenos *clusters* fazendo a média da distribuição se aproximar deles e, ao mesmo tempo, se afastar das observações normais dificultando a identificação dos verdadeiros *outliers*.

$$z\_score = \frac{|x_i - \mu|}{\sigma} \tag{3.1}$$

Outra técnica para dados univariados foi proposta por Davies e Gather (1993) sendo definida da seguinte maneira. Dada uma amostra $X_N$ de um conjunto de dados de distribuição normal $N(\mu, \sigma^2)$ com os parâmetros $\mu$ representando a média e $\sigma$ o desvio padrão, o problema de detecção de *outliers* se resume em identificar a região *outlier* definida por $out(\alpha, \mu, \sigma^2) = \{x : |x - \mu| > z_{1-\alpha/2^\sigma}\}$, onde $\alpha \in (0, 1)$ é o nível de significância especificado por um estatístico e $z_q$ é o $q - quantil$ da distribuição normal $N(0, 1)$. Para um limiar inferior $L(X_N, \alpha_N)$ e um limiar superior $R(X_N, \alpha_N)$ definido pela amostra e pelo parâmetro $\alpha$, os dados menores que $L(X_N, \alpha_N)$ ou maiores que $R(X_N, \alpha_N)$ são declarados $\alpha$ *outliers* pois pertencem a região *outlier* $out(\alpha, \mu, \sigma^2)$ com nível de significância $\alpha$.

Testes de hipótese podem ser aplicados para detecção de *outliers*. São estabelecidas a hipótese que os dados foram gerados usando uma determinada distribuição e uma hipótese alterna-

tiva que é oposta à hipótese de interesse. Rejeitar a hipótese de interesse significa que o dado foi gerado por uma distribuição diferente. A escolha do teste de discordância depende do conhecimento da distribuição dos dados, dos parâmetros, quantidade e tipo de *outliers* esperados (Agyemang et al., 2006).

Em conjuntos de dados multivariados com distribuição normal, uma forma de detectar *outliers* é por meio da distância de Mahalanobis, descrita na seção 2.1.2, em que os dados com grande distância são considerados *outliers*. Apesar de simples, a técnica é custosa para conjuntos de dados com altas dimensões devido à necessidade do cálculo da matriz de covariância. Aliás, bases de dados de alta dimensão tipicamente são geradas por várias distribuições diferentes que são desconhecidas dificultando a estimação dos parâmetros das distribuições e a construção de testes de hipótese. Técnicas de redução de dimensionalidade como *PCA* (*Principle Component Analysis*) e *MDS* (*Multi Dimensional Scaling*) tentam mitigar o problema do custo computacional através da projeção dos dados em um subespaço de menor dimensão onde os *outliers* estão em regiões de baixa densidade.

As técnicas estatísticas exibem uma solução adequada quando a distribuição dos dados é conhecida mesmo se os rótulos dos dados não estão disponíveis. Entretanto, é preciso fornecer os parâmetros da distribuição ou estimar o modelo estatístico a partir dos dados. O primeiro caso refere-se às técnicas estatísticas paramétricas baseadas em modelos gaussianos, modelos de regressão ou modelos de mistura nos quais os testes de discordância são usados com frequência. Já as técnicas estatísticas não paramétricas são convenientes para dados que não seguem uma distribuição específica.

### 3.2.2 Técnicas baseadas em distância

Técnicas baseadas em distância são aquelas que variam a forma de computar a distância de um objeto em relação aos k-vizinhos mais próximos. Em termos de complexidade de tempo, possuem ordem quadrática em relação ao tempo computacional, pois dependem do cálculo dos vizinhos mais próximos. Knorr e Ng (1997) propuseram a seguinte definição de *outlier*:

**Definição:** DB *outlier* é um objeto que está a uma distância maior que *D* de pelo menos *p*% de outros objetos.

Do ponto de vista global, um objeto é anormal se estiver distante da maioria dos objetos. Os parâmetros *D* e *p* são definidos pelo usuário e restringem a quantidade de *outliers* encontrados. A escolha dos valores desses parâmetros depende de uma série de fatores como, por exemplo, quão esparsos os objetos estão. Essa definição é capaz de identificar apenas alguns tipos de *outliers* apresentando problemas quando o conjunto de dados tem diferentes densidades ou contém *clusters* de tamanhos diferentes.

Outra definição foi dada por Ramaswamy et al. (2000) para medir o grau de anormalidade de um objeto baseado na distância entre vizinhos mais próximos. Seja $k$ o número de vizinhos mais próximos, $D^k$ a distância de um objeto até seu $k$-ésimo vizinho mais próximo e $n$ o total de *outliers* esperado.

**Definição:** Um objeto é um $D_n^k$ *outlier* com respeito aos parâmetros $k$ e $n$, se não existe mais que $n-1$ objetos $p'$ tal que $D_n^k(p') > D_n^k(p)$.

Assim, os $n$ objetos com as maiores distâncias ao $k$-ésimo vizinho mais próximo são potenciais *outliers*. Se mais que n objetos satisfizerem a definição, eles também são considerados $D_n^k$ *outliers*. Assume-se que $n$ tenha um valor pequeno e não dependa, de certa forma, do conjunto de dados (Ramaswamy et al., 2000). A distância pode ser medida por meio da distância de Manhattan ou Euclidiana e a escolha do valor não depende do conjunto de dados.

Visto que o cálculo dos $k$ vizinhos mais próximos tem complexidade de ordem quadrática em relação ao número de objetos, modificações têm sido elaboradas para melhorar o desempenho. Uma ideia é dividir o espaço vetorial em retângulos e podar aqueles retângulos que possuem apenas dados normais. Ramaswamy et al. (2000) desenvolveram uma técnica composta por etapas isoladas. O primeiro passo é utilizar um agrupamento para gerar as partições. Em seguida, calcula-se os limites inferiores e superiores para $D^k$ de cada partição. Partições cujas $D^k$ são maiores que $p$ são podadas e as partições restantes são analisadas na busca por *outliers*.

Existem técnicas que se baseiam no cálculo da densidade de objetos para identificar *outliers*. Consideram que um dado normal está localizado em uma região densa de objetos e um *outlier* está em uma região de baixa densidade. $LOF$ (*Local Outlier Factor*)(Breunig et al., 2000) é uma técnica de detecção de *outlier* que atribui um fator de anormalidade local para cada dado computado pela diferença da densidade de um objeto em relação a sua vizinhança. Dado o parâmetro de entrada ***MinPts*** que corresponde à quantidade mínima de objetos no cálculo da vizinhança de um objeto e d(*p*,*o*) a distância entre dois objetos *p* e *o*, define-se as seguintes variáveis:

$MinPts\_distance(p)$**:** distância entre um objeto *p* e seu $MinPts\_vizinhos$ mais próximos. Pode ser entendida como o raio da menor esfera, centrada em *p*, que inclui os ***MinPts*** objetos mais próximos de *p*.

$N_{MinPts}(p) = \{q \in D \text{ e } q \neq p \mid d(p,q) \leq MinPts\_distance(p)\}$, isto é, o conjunto formado pelos $MinPts\_vizinhos$ mais próximos de *p*. Note que a cardinalidade desse conjunto pode ser maior que ***MinPts***.

$reach\_dist\_MinPts(p,o) = max\{MinPts\_distance(o), d(p,o)\}$. O valor é igual a $d(p,o)$ se o objeto *p* não está na vizinhança do objeto *o* ou igual a $MinPts\_distance(o)$ caso contrário.

Define-se como ***Local Reachability Density*** de um objeto *p* a equação

$$lrd_{MinPts}(p) = \frac{1}{\frac{\sum_{o \in N_{MinPts(p)}} reach\_dist_{MinPts}(p,o)}{|N_{MinPts}(p)|}} \tag{3.2}$$

em que $|N_{MinPts}(p)|$ representa a cardinalidade do conjunto. Resumidamente, essa equação mede a densidade de um objeto *p*. Quando o valor de *lrd* é próximo de 1 indica que os objetos

vizinhos de *p* estão próximos, ou seja, *p* está em uma região de alta densidade. Se o valor é próximo de zero, os objetos estão afastados indicando que *p* está em uma região de baixa densidade.

O ***Local Outlier Factor*** de um objeto *p* é dado por

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts(p)}} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \tag{3.3}$$

e calcula a média da razão entre a densidade dos objetos vizinhos de *p* e a densidade do objeto *p*. Novamente, se um objeto *o* está em uma região de alta densidade e o objeto *p* está em uma região de baixa densidade, a fração $\frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}$ terá valor maior que 1 contribuindo para o aumento do $LOF$. A magnitude do $LOF$ para um objeto *outlier* depende da densidade do *cluster* mais próximo e da distância desse objeto até esse *cluster*. Além disso, os objetos pertencentes a um *cluster* possuem $LOF$ aproximadamente igual a 1 (Breunig et al., 2000).

Apesar de obter bons resultados na identificação de objetos isolados, a técnica $LOF$ tem algumas desvantagens como complexidade computacional alta devido à necessidade de calcular as distâncias entre cada par de objetos e ineficiência para identificar pequenos *clusters* de *outliers*. Em (Jin et al., 2001), é proposta uma técnica para encontrar apenas os $top - n$ *outliers* locais usando uma representação resumida de cada *cluster* (*micro-cluster*). Outro detalhe importante é que a técnica $LOF$ não é adequada para detectar *outliers* que ocorrem em regiões de alta densidade dado que os objetos da classe normal formam padrões regulares e de baixa densidade (Hu e Sung, 2003).

Hubballi et al. (2011) propuseram uma técnica chamada de *NDoT* (*Nearest Neighbor Distance Based outlier Detection Technique*) que classifica dados como *outliers* através de um mecanismo de votação (Algoritmo 3). O primeiro passo é determinar o conjunto de vizinhos mais próximos $NN_p$ de cada objeto $p$. Em seguida, é calculada a distância média $meandist_{NN}(p)$ do objeto $p$ (equação 3.4) em relação a todos os seus vizinhos mais próximos considerando $d(p, o)$ como a distância euclidiana entre os objetos $p$ e $o$. Então é proposto um fator $NNF(p, o)$ (equação 3.5) do objeto $p$ em relação ao objeto $o$ obtido pela razão entre a distância $d(p, o)$ e a distância média $meandist_{NN}(o)$. Quando o valor deste fator é superior ou igual ao valor de um parâmetro $\delta$, o número de votos $V_p$ do objeto $p$ é incrementado em 1. Um objeto é classificado como *outlier* se receber mais de $\frac{2}{3}$ de votos de seus vizinhos indicando que ele não está em conformidade com seus vizinhos mais próximos. Simulações realizadas pelos autores demonstraram que *NDoT* é comparável à técnica *LOF*.

**Algoritmo 3** Algoritmo NDoT.

**Entrada:** conjunto de dados $D = \{p_1, p_2, \ldots, p_n\}$, parâmetros $\delta$ e $k$.
Obtém o conjunto $NN_p$ formado pelos seus $k$ vizinhos mais próximos de cada objeto $p \in D$.

Calcula a distância média de cada objeto $p$ aos seus vizinhos mais próximos (equação 3.4).
**para todo** $p \in D$ **faça**
  $V_p = 0$.
  **para todo** $o \in NN_p$ **faça**
    **se** $NNF(p, o) \geq \delta$ **então**
      $\{NNF(p, o)$ é obtido pela equação 3.5$\}$
      O objeto $p$ recebe um voto, isto é, $V_p = V_p + 1$.
    **fim se**
  **fim para**
  **se** $V_p \geq \frac{2}{3}|NN_p|$ **então**
    Classifica o objeto $p$ como *outlier*.
  **fim se**
**fim para**

$$meandist_{NN}(p) = \frac{\sum\limits_{o \in NN_p} d(p, o)}{|NN_p|} \tag{3.4}$$

$$NNF(p, o) = \frac{d(p, o)}{meandist_{NN}(o)} \tag{3.5}$$

### 3.2.3 Técnicas baseadas em agrupamento de dados

O processo chamado de agrupamento ou clusterização de dados consiste em dividir os dados em *clusters* tal que dados similares pertençam ao mesmo *cluster*. Algumas classes de algoritmos de agrupamento são discutidas na seção 2.3.4. Cada classe tem uma forma específica de manipulação dos dados e os problemas mais gerais incluem a seleção de uma função apropriada para avaliação do agrupamento, escolha do algoritmo e número de *clusters* gerados pelo agrupamento.

Sabe-se que *outliers* podem interferir no processo de agrupamento de dados, pois quando estão isolados não se enquadram em nenhum *cluster*. Apesar de alguns algoritmos de agrupamento de dados serem capazes de tratar essa questão, eles se concentram na identificação de *clusters* e não na detecção de *outliers*. Então, o resultado da detecção é fortemente influenciado pelo pressuposto usado na definição de *clusters*, isto é, necessita-se estabelecer algum critério em relação a distribuição dos dados que caracterize a diferença entre um dado normal e um dado anormal. A seguir, serão descritas duas categorias de técnicas baseadas em agrupamento de dados.

Na primeira categoria objetos normais pertencem a algum *cluster* e os *outliers*, considerados como ruídos, não pertencem a nenhum *cluster*. Exemplos de algoritmos dessa categoria são: *DBSCAN* (Ester et al., 1996) e *BIRCH* (Zhang et al., 1997). O *BIRCH* considera como *outliers* ou ruídos as entradas de baixa densidade presentes nos nós folhas da árvore CF.

*SSOD* é uma técnica descrita em (Gao et al., 2006), que faz uso do algoritmo de agrupamento *K-Médias* para separar dados normais de *outliers*. Ela segue a abordagem semissupervisionada, pois inclui dados rotulados e não rotulados na etapa de treinamento do modelo. Para um conjunto $D = \{x_1, x_2, \ldots, x_n\}$ com $n$ instâncias, $K$ *clusters* e o vetor $\mu = \{\mu_1, \mu_2, \ldots, \mu_l\}$ contendo os rótulos das $l$ primeiras instâncias define-se uma matriz $T = \{t_{ih} | 1 \leq i \leq n, 1 \leq h \leq K\}$ similar à matriz do algoritmo *K-Médias* tradicional, mas com a restrição

$$\sum_{h=1}^{K} t_{ih} = \begin{cases} 1, & \text{se} \quad x_i \quad \text{é um ponto normal,} \\ 0, & \text{se} \quad x_i \quad \text{é um } \textit{outlier}. \end{cases} \tag{3.6}$$

A matriz $T$ é obtida pela minimização da função objetivo $Q$ descrita na equação 3.7, a qual é composta pela soma de três expressões. A primeira expressão computa a distância dos objetos normais ao centróide mais próximo, a segunda expressão adiciona uma constante $\gamma_1$ para cada objeto declarado como *outlier* e a terceira expressão adiciona uma constante $\gamma_2$ para cada objeto classificado incorretamente. Uma interpretação do parâmetro $\gamma_1$ é dada pela Figura 3.4. Ele pode ser visto como a distância máxima permitida que um dado pode ter em relação ao centróide mais próximo para ser considerado como normal e também está relacionado com a quantidade de *outliers* encontrados. Em cada passo, além da minimização da função objetivo $Q$, a posição $c_h$ de cada centróide $h$ é atualizada em cada iteração $s$ levando em conta apenas os objetos normais como mostra a equação 3.8.

$$Q = \sum_{i=1}^{n}\sum_{h=1}^{K} t_{ih} dist(c_h, x_i)^2 + \gamma_1 (n - \sum_{i=1}^{n}\sum_{h=1}^{K} t_{ih}) + \gamma_2 \sum_{i=1}^{l} \mid \mu_i - \sum_{h=1}^{K} t_{ih} \mid \tag{3.7}$$

$$c_h^{s+1} = \frac{\sum_{i=1}^{n}(t_{ih} x_i)}{\sum_{i=1}^{n} t_{ih}} \tag{3.8}$$

Uma estratégia similar, mas que não faz uso de exemplos rotulados, é usada no *Noise Clustering* (Oliveira e Pedrycz, 2007). Dadas as matrizes de distância dos dados aos *clusters* e do grau de pertinência dos dados a cada um dos *clusters*, geradas por um algoritmo de agrupamento *fuzzy*, a função de minimização possui um termo de penalidade para que dados com baixa representatividade sejam adicionados a um *cluster de ruído*. Os *outliers* apresentarão alto grau de pertinência ao *cluster de ruído* e baixo grau de pertinência aos *clusters* normais.

Outro algoritmo de agrupamento adaptado para tratar desse problema é o *Wave Cluster* (Sheikholeslami et al., 1998). Utilizando a *transformada de wavelets*, ele foi desenvolvido para agrupamento de dados espaciais de grandes bases de dados de baixa dimensão. Yu et al. (2002) propuseram uma técnica que realiza a decomposição multi-resolução de sinais baseada em *wavelet*. A transformada de *wavelet* é aplicada nos dados para convertê-los no domínio de frequências composto por sinais d-dimensionais. O conjunto de dados é composto por sinais cujas partes de alta frequência indicam fronteiras entre *clusters* devido a mudança brusca da

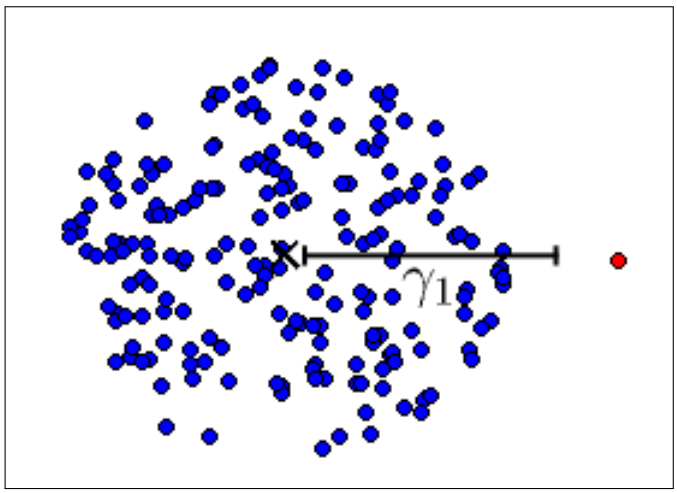



**Figura 3.4:** Interpretação do parâmetro $\gamma_1$ da técnica *SSOD* considerando um *cluster* esférico com o centróide representado pelo simbolo $X$. Esse parâmetro representa a distância máxima permitida em relação ao centróide para que um dado seja considerado como normal. O ponto mais distante dos demais, indicado pela cor vermelha, é considerado um *outlier* pois sua distância até o centróide mais próximo tem valor superior a $\gamma_1$.

distribuição dos dados. As partes de baixa frequência e alta amplitude correspondem a regiões com alta concentração dos dados. Dessa forma, supõem que *outliers* estão localizados em áreas de baixa densidade.

A segunda categoria de técnicas baseadas em agrupamento de dados supõe que objetos normais pertencem a *clusters* grandes e densos enquanto *outliers* pertencem a *clusters* pequenos e esparsos. Isso significa que *outliers* não são mais vistos como dados isolados. He et al. (2003) criaram uma medida chamada *cluster-based local outlier factor* (*CBLOF*) para identificar *clusters outliers*. Seja $C = \{C_1, C_2, ..., C_k\}$ um conjunto de $k$ clusters, em ordem decrescente de tamanho, gerado por um algoritmo de agrupamento particional tal que $C_i \cap C_j = \emptyset$ e $C_1 \cup C_2 \cup ... \cup C_k = D$. Dados os parâmetros numéricos $\alpha$ e $\beta$, os primeiros b *clusters* são considerados grandes (LC) e os demais são considerados pequenos (SC) conforme as inequações 3.9 e 3.10. Os *clusters* grandes devem conter a maior parte dos dados (inequação 3.9) e a diferença de tamanho entre o menor *cluster* grande e o maior *cluster* pequeno deve ser significativa (inequação 3.10).

$$(|C_1| + |C_2| + ... + |C_b|) \geq |D| * \alpha \tag{3.9}$$

$$\frac{|C_b|}{|C_{b+1}|} \geq \beta \tag{3.10}$$

Para um objeto *p* do conjunto de dados, o *CBLOF* é calculado pela equação 3.11 na qual $dist(p, C_j)$ é a distância do objeto *p* ao centróide do *cluster* $C_j$. He et al. (2003) usam a medida *CBLOF* juntamente como algoritmo de agrupamento de dados categóricos *Squeezer* para identificar *clusters outliers*.

$$CBLOF(p) = \begin{cases} |C_i| * min\Big(dist(p, C_j)\Big), & \text{se } p \in C_i, C_i \in SC \\ & \text{e } C_j \in LC \text{ para j de 1 até b} \\ \\ |C_i| * dist(p, C_i), & \text{se } p \in C_i, C_i \in LC \end{cases} \tag{3.11}$$

É possível observar que técnicas da segunda categoria são influenciadas pelo número de *clusters* encontrados. Apesar de ser difícil atribuir grau de anormalidade para o objeto seguindo esse esquema, a diferença absoluta entre a qualidade do agrupamento antes e depois da remoção de um objeto pode ser usada como grau de anormalidade.

### 3.2.4 Técnicas baseadas em redes

Técnicas baseadas em redes são aquelas que misturam estratégias empregadas por outras categorias e representam o conjunto de dados na forma de rede. Segundo Eberle e Holder (2007), os *outliers* são pequenas mudanças em relação ao padrão normal categorizados em anomalias de inserção, modificação ou remoção. A anomalia de inserção refere-se à existência inesperada de um vértice ou aresta na rede. O segundo caso ocorre quando um vértice ou aresta possui um rótulo diferente do esperado. Já a anomalia de remoção é caracterizada pela ausência de um vértice ou aresta esperada na rede.

Os seguintes tópicos estão relacionados à detecção de *outliers* em redes:

- Vértices com características incomuns;
- Arestas com características incomuns: arestas mais importantes ou estatisticamente improváveis. Outra possibilidade é a utilização de técnicas de previsão de arestas (*anomaly link discovery*) (Rattigan e Jensen, 2005; Huang e Zeng, 2006).
- Detecção de subredes anormais. A identificação de *motifs*. Um pré-requisito para detecção dessas subredes é entender os padrões que ocorrem frequentemente (Cook e Holder, 2006; Eberle e Holder, 2007).
- Identificar evolução fora do padrão esperado de uma rede que muda ao longo do tempo (Chen et al., 2011).
- Comparação da rede com um conjunto de redes geradas aleatoriamente. Quanto maior for a diferença entre eles, mais *outlier* a rede é.
- Busca por simplicidade em redes. Redes mais simples são aquelas que possuem vértices com mesmas características.

*Outrank* (Moonesignhe e Tan, 2006) é uma técnica baseada na caminhada de *Markov* e usa a similaridade do cosseno para atribuir peso na conexão entre dois objetos $X$ e $Y$ quaisquer con-

forme mostra a equação 3.12. Nesta equação $x_k$ e $y_k$ representam os valores de um determinado atributo e $m$ indica a dimensão do conjunto de dados.

$$S(X,Y) = \begin{cases} 0, & \text{se} \quad \text{X = Y,} \\ \frac{\sum_{k=1}^{m} x_k y_k}{\sqrt{\sum_{k=1}^{m} x_k}\sqrt{\sum_{k=1}^{m} y_k}}, & \text{caso contrário.} \end{cases} \tag{3.12}$$

A matriz de similaridade, denotada por $S$, é normalizada a fim de satisfazer as propriedades da *matriz de Markov*. Quando todos os elementos dessa matriz forem positivos e os valores de cada coluna somam 1, o maior autovalor da *matriz de Markov* é 1 (Strang, 2009). Logo, o autovetor associado ao maior autovalor é um estado estacionário. Seja $d$ um fator para garantir que a matriz $S$ seja irredutível e aperiódica, a distribuição estacionária $c$ que representa a conectividade dos vértices é dada por:

$$c = d + (1 - d)Sc. \tag{3.13}$$

Após um número suficiente de iterações a distribuição estacionária $c$ é atingida independente do valor inicial do vetor $c$ desde que a soma de seus elementos seja $1$. Cada elemento de $c$ indica o grau de conectividade de um vértice. Vértices com baixa conectividade são potenciais *outliers*.

Como a simples análise do grau de um vértice não ajuda a caracterizar a rede como um todo, em (Berton et al., 2010) é calculado um *score* para cada vértice da rede utilizando a medida de distância de uma caminhada aleatória. Seja $p_{ij}$ a probabilidade de estar no vértice $j$ no estado estacionário partindo do vértice $i$. Considerando uma rede com $n$ vértices, o índice de dissimilaridade $\Lambda$, descrito em (Zhou, 2003), entre os vértices $i$ e $j$ é obtido pela equação 3.14. A matriz simétrica obtida é usada na função $\sigma$ que avalia a perspectiva de cada vértice em relação aos demais (equação 3.15). Vértices com os maiores valores são os potenciais *outliers*.

$$\Lambda(i,j) = \frac{\sqrt{\sum_{k \neq i,j}^{n} (p_{ik} - p_{jk})^2}}{(n-2)} \tag{3.14}$$

$$\sigma(i) = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \Lambda(i,j) \tag{3.15}$$

Em (Hautamaki et al., 2004) são descritas duas técnicas de detecção de *outliers* baseadas na rede $kNN$. O primeiro, chamado de *MeanDIST algorithm*, calcula a média dos pesos das arestas de cada vértice compondo um vetor $L$ em ordem crescente desses valores. Defini-se um limiar $t$ e calcula-se a diferença entre valores vizinhos conforme a equação 3.16. Todos os vértices com índice maior ou igual a $i$ tal que $L_i - L_{i-1} \geq T$ são considerados *outliers*.

$$T = max(L_i - L_{i-1}) * t \tag{3.16}$$

A segunda técnica, chamada ODIN (*Outlier Detection using Indegree Number*), computa vértices *outliers* pelo grau de entrada de cada vértice na construção da rede $kNN$ direcionada e ponderada do conjunto de dados. O grau de entrada de um vértice $v_i$ corresponde a quantidade de vértices que possuem o vértice $v_i$ como vizinho. Então, um vértice *outlier* é aquele que possui grau de entrada menor que um parâmetro $T$ pré-definido.

Todas as técnicas apresentadas anteriormente dessa categoria detectam vértices *outliers*. Uma das formas proposta por Noble e Cook (2003) para detectar subredes anormais é examinar o número de ocorrências das subestruturas. As subredes com subestruturas comuns são menos anormais que aqueles com subestruturas incomuns. Após detectar as subestruturas mais frequentes em uma rede, é medida a frequência de uma subrede $S$ na rede $G$ pela equação 3.17. $Size(S)$ retorna o número de vértices da subrede $S$ e $Instances(S, G)$ retorna o número de vezes que a subrede $S$ aparece na rede $G$. As subestruturas mais anormais são aquelas que retornam um valor pequeno de $F2$. Esse esquema impede que a rede completa ou que um vértice isolado tenha alto grau de anormalidade.

$$F2(S, G) = Size(S) * Instances(S, G) \tag{3.17}$$

Apesar de haver muitos trabalhos nesta área, poucos deles focam na detecção de *outliers* em redes. Um dos obstáculos que ainda se mantém é a falta de uma definição formal de *outlier* em rede. Noble e Cook (2003) focaram na identificação de subredes incomuns. Moonesignhe e Tan (2006) consideram que um vértice com baixa conectividade tem alta probabilidade de ser um *outlier*, embora esse resultado seja influenciado pelo método de geração de rede. Neste último caso, vértices que apresentam um grau muito longe da média são candidatos a *outliers*. Em uma rede de Internet, por exemplo, poucos vértices apresentam muitas ligações e geralmente desempenham funções diferentes dos demais. Esses vértices, chamados de *hubs*, são responsáveis pelo roteamento de grande parte da informação que trafega na rede. Outras abordagens baseadas em redes são encontradas em (Shekhar et al., 2001) e (Costa et al., 2009)

### 3.2.5 Técnicas baseadas na teoria da informação

Uma suposição feita por técnicas baseadas na teoria da informação é que existe regularidade no conjunto de dados, isto é, certas características ou instâncias ocorrem com mais frequência em relação a outras. Isso significa que o conjunto de dados não pode ter uma distribuição uniforme já que todos os dados dessa distribuição tem mesma probabilidade de ocorrência. A regularidade pode ser medida, por exemplo, através da diferença entre a entropia do conjunto e a entropia da distribuição aleatória (Pan e Wang, 2006). Considerando que a distribuição aleatória não admite regularidade (possui alta entropia), o valor obtido daquela diferença indica o quão regular um conjunto de dados é. A partir dessa ideia é possível estimar a regularidade dos dados que exibem um comportamento normal. Quando um *outlier* é inserido neste conjunto, haverá um aumento significativo na entropia do conjunto indicando que esse dado apresenta um comportamento diferenciado.

Shetty e Adibi (2005) assumem que vértices mais importantes são aqueles que causam um efeito maior na entropia da rede quando removidos. Seguindo essa estratégia, eles analisaram uma base de dados de emails com objetivo de identificar as pessoas mais importantes de uma organização. Com base nos resultados obtidos, percebeu-se que os vértices mais importantes ou anormais da rede não são necessariamente os vértices centrais.

Pode ocorrer que o valor da entropia seja igual para dois conjuntos de dados com diferentes distribuições. Isso seria um problema, por exemplo, para distinguir o comportamento de conjuntos de atributos. Neste caso, uma possível solução seria a medida *Kullback-Leibler* que calcula a entropia relativa entre duas distribuições (Quan et al., 2009). No problema de detecção de intrusão, ataques podem ser caracterizados por um número grande de requisições tal como o ataque por negação de serviço. Situações como essa são propícias para utilização de medidas derivadas da entropia relativa.

### 3.2.6 Outras técnicas de detecção de outliers

Redes neurais são usualmente empregadas em aplicações de detecção de fraudes de cartão de crédito, detecção de intrusão e detecção de fraudes em telecomunicações. Dependendo do tipo de rede neural, a detecção de fraudes é feita por meio de um classificador, gerado com base no conjunto dados normais da fase de treinamento. Na fase de teste, padrões reconhecidos são considerados como normais enquanto os padrões com alto erro de reconstrução são ditos *outliers*. O erro quadrático médio pode ser usado como medida de grau de anormalidade. S. Hawkins et al. (2002) usam uma rede neural *multi-layer perceptron feed-forward* com $d$ entradas e $d$ saídas, e definem o grau de anormalidade (*OF*) de um objeto $x_i$ como

$$OF_i = \frac{1}{d}\sum_{j=1}^{d}(x_{ij} - o_{ij}^{l})^2 \,, \tag{3.18}$$

em que $x_{ij}$ é o atributo $j$ do objeto $x_i$ apresentado na entrada da rede e $o_{ij}^{l}$ é o valor do atributo $j$ obtido na saída da rede na iteração $l$.

Técnicas baseadas em profundidade definem, como o próprio nome sugere, uma profundidade para cada dado representado no espaço. Em um espaço bidimensional, a profundidade de um ponto é definida, por exemplo, pela quantidade mínima de dados contidos em um semi plano fechado limitado por uma linha que passa por esse ponto (Agyemang et al., 2006). Não requerem conhecimento da distribuição dos dados, mas geralmente são ineficientes computacionalmente para tratar de dados em alta dimensão porque a definição do conceito de profundidade e volume é complexa (Agyemang et al., 2006).

Outra possibilidade é particionar o conjunto de dados em células e fazer uso de estruturas de indexação para auxiliar na tarefa de detecção de *outliers*. Todavia, tem sido um tópico pouco explorado em razão do processo de particionamento do conjunto de dados produzir muitas células vazias quando os dados são esparsos ou quando a dimensão é alta prejudicando o desempenho do algoritmo. H. Sun et al. (2004) utilizam uma estrutura chamada de *CD-Tree* (*Cell Dimension Tree*) para detecção de *outliers*. As células da *CD-Tree* são retângulos não sobrepostos formadas

pela intersecção de um intervalo de cada atributo. Outra característica da *CD-Tree* é de não armazenar células vazias. Para tanto, a estrutura é originada do conceito de *SOD* (*Skew Of Data*), usado para estimar a porção de células vazias de cada partição. Dados candidatos a serem *outliers* possuem poucos dados em sua vizinhança ou estão em células que contêm poucos dados. Chaudhary et al. (2002) dividem o espaço em células compostas por hiper-retângulos com o auxílio da *KD-Tree* e analisam a esparsividade, inverso da densidade, de cada região como grau de anormalidade (*outlierability*). A árvore *KD-Tree* é usada para consultar a localização dos dados, consultar os vizinhos mais próximos e detectar *outliers* mais rapidamente. Cada célula pode receber cortes especiais se os dados estão em uma região irregularmente distribuída, isto é, existe pelo menos uma sub-região densa e uma sub-região esparsa. Depois, os dados de cada célula folha são ranqueados usando um fator de suavidade (Chaudhary et al., 2002) e os *outliers* são considerados como desvios que aumentam a dissimilaridade do conjunto de dados.

As limitações encontradas nas categorias convencionais têm instigado pesquisadores a desenvolver alternativas híbridas. O foco principal continua sendo obter alta precisão e baixo alarme falso. Este último requisito é exigido para técnicas que atribuem rótulos aos objetos.

## 3.3 Considerações Finais

Neste capítulo, apresentou-se uma revisão geral sobre detecção de *outliers* abordando os aspectos gerais desses dados e algumas das principais categorias de técnicas desenvolvidas para esse problema. Destacou-se as vantagens e desvantagens de cada categoria, além da suposição adotada por elas. Existem ainda outras categorias não detalhadas neste capítulo como aquelas que tratam de dados espaciais (Kou e Lu, 2006; Shekhar et al., 2001) e aquelas vinculadas a termos da web. Esta última é voltada para análise de dados de web como *hyperlinks*, hipertexto, vídeo, palavras-chave, etc.

Foi visto que a detecção de *outliers* não é uma tarefa trivial por diversos motivos. O primeiro deles é devido à dificuldade em definir, com precisão, a fronteira entre dados normais e *outliers*. Em certos domínios, a noção de comportamento normal pode variar ao longo do tempo necessitando que o processo de treinamento seja realizado periodicamente. Outro motivo refere-se ao problema do aumento no tempo de processamento e da maldição da dimensionalidade quando manipula-se *outliers* em conjunto de dados de alta dimensionalidade. O aumento do número de dimensões deixa os dados mais esparsos, de modo que a diferença entre a distância de pontos próximos e distantes torna-se menos evidente. Em geral, as técnicas devem ser capazes de manipular dados com atributos simbólicos e não devem depender do ajuste de parâmetros pelo usuário.

Capítulo
4

# Técnica de detecção de outliers baseada em competição e cooperação de partículas

Detectar *outliers* é uma importante tarefa para extração de conhecimento. Apesar de existirem várias técnicas de detecção de *outliers*, há uma carência de técnicas baseadas em redes. Redes são representações flexíveis e poderosas já que as relações entre os dados ficam mais evidentes em comparação com a representação na forma de atributo-valor. Outro ponto importante é que redes não servem apenas para representar a similaridade entre os dados, mas também auxiliam o descobrimento de padrões estruturais formados pelos dados de entrada. Como muitas técnicas baseadas em redes têm obtido bons resultados em reconhecimento de padrões, podem ser adequadas também para identificar objetos fora do padrão.

Este capítulo descreve uma nova técnica de detecção de *outliers* utilizando o modelo semissupervisionado de competição e cooperação de partículas descrito no Capítulo 2. A técnica consiste em uma medida de caracterização de vértices da rede usando as informações provenientes do processo de competição e cooperação de partículas. Por se tratar de uma técnica baseada em rede, as bases de dados na forma de tabela atributo-valor deverão ser convertidas em rede. Algumas formas de construção de redes são apresentadas na seção 2.4.5. O tópico de técnicas de construção de redes que representem adequadamente os conjunto de dados é um assunto além do escopo deste trabalho. Sabe-se que essas técnicas dependem de uma boa construção da rede para produzirem bons resultados.

Na primeira seção, são descritas as modificações realizadas no modelo original, que incluem a definição de um parâmetro e de um escore de *outlier*. Na seção seguinte, são apresentadas simulações realizadas em bases de dados artificiais com intuito de ilustrar o comportamento da

técnica. Realizou-se também uma comparação entre a técnica proposta e as seguintes técnicas de detecção de *outliers*: k-médias semissupervisionado para detecção de outliers (*SSOD*) (Gao et al., 2006), *outrank-b* (Moonesignhe e Tan, 2006), *NDOT* (Hubballi et al., 2011) e *Local Outlier Factor* (Breunig et al., 2000). Algumas dessas técnicas são bem conhecidas e estão descritas no Capítulo 3.2. As próximas seções apresentam análises da eficiência técnica proposta em relação aos parâmetros, quantidade de exemplos rotulados, quantidade de *outliers* e influência da rede. No final do capítulo são descritas as considerações finais.

## 4.1 Medida de detecção de outliers baseada em frequência de visitas

O modelo semissupervisionado de competição e cooperação de partículas (Breve et al., 2011; Silva e Zhao, 2012) foi originalmente proposto para classificação de dados através da propagação de rótulos. Essa propagação é realizada por partículas as quais são separadas em times de acordo com o número de classes existentes e são responsáveis por transmitir o rótulo da classe que ela está vinculada. Partículas de mesmo time cooperam entre si para disseminar o rótulo de sua classe e competem com outras de diferentes times na tentativa de conquistar o maior número possível de vértices. Após a convergência do algoritmo, cada vértice recebe o rótulo do time de partículas que o domina. O modelo original está descrito em maiores detalhes na seção 2.4.5.

A motivação para o uso desse modelo partiu dos resultados apresentados em (Silva e Zhao, 2012; Breve et al., 2011). Segundo os autores, o modelo tem habilidade para classificar bases de dados que contêm *clusters* de formas arbitrárias utilizando poucos dados rotulados. Ao contrário da maioria dos modelos baseados em redes, ele tem baixo custo computacional. A ordem de complexidade é no máximo quadrática em relação ao número de vértices enquanto a maioria dos modelos tem de ordem de complexidade cúbica. Os autores também estimam os valores adequados para os parâmetros.

Na literatura, a detecção de *outliers* é comumente tratada como um problema de análise da densidade dos objetos. Entretanto, técnicas baseadas em densidade não são adequadas para bases que contêm padrões de baixa densidade. Por sua vez, técnicas baseadas em redes são capazes de identificar estruturas complexas e não convencionais. Essa capacidade motivou a utilização de um modelo baseado em rede para detecção de *outliers*. Procurou-se, então, definir um escore de *outlier* usando as informações provenientes do processo de competição e cooperação de partículas.

Visto que o modelo original funciona sobre uma base de dados na forma de rede, por padrão adotou-se a técnica de vizinhos mais próximos ($k$NN) para geração da rede, salvo quando o processo de geração da rede estiver descrito. A abordagem $k$NN foi utilizada para construir uma rede não direcionada e ponderada onde cada vértice está conectado com os $k$ vértices mais similares em $\mathcal{V}$. Seja $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ uma base de dados com $n$ objetos, define-se

$NNk(x_i) = \{x_j \in \mathcal{X} \mid d(x_i, x_j) \leq d(x_i, x_j^{'}), x_j^{'} \in \mathcal{X}\}$ como o conjunto de vizinhos mais próximos do objeto $x_i$. Então, a matriz de similaridade é dada por

$$W_{ij} = \begin{cases} \frac{1}{d(x_i,x_j)}, & \text{se} \quad x_j \in NNk(x_i) \\ \\ 0, & \text{se} \quad x_j \notin NNk(x_i). \end{cases} \tag{4.1}$$

A primeira modificação do modelo consiste em armazenar o caminho percorrido por cada partícula. Para tanto, foi incorporado um parâmetro $\tau$ que define o tamanho da lista de vértices visitados, isto é, o caminho percorrido por cada partícula nas $\tau$ iterações anteriores. A lista de cada partícula é uma fila que é atualizada removendo o vértice do topo da fila e adicionando o vértice recentemente visitado no final da fila. O objetivo é impedir que uma partícula visite os vértices que estejam nesta lista. Esta regra não se aplica ao vértice que a partícula ocupa. A hipótese é que essa abordagem tende a favorecer a detecção de pequenos grupos de *outliers*, pois evita que as partículas realizem ciclos pequenos. Sem a inclusão dessa lista, uma partícula poderia permanecer por um longo tempo dentro de um *cluster outlier* e, dessa forma, os *outliers* receberiam muitas visitas. Embora a inclusão dessa lista não impeça que partículas visitem *clusters outliers*, ela direciona o caminho para outras regiões.

O esquema de funcionamento da lista é ilustrado na Figura 4.1. Suponha que uma partícula, ilustrada por um círculo preenchido com cor preta, inicie sua caminhada a partir do vértice $v_1$ no instante $t$ e que $\tau = 4$. Nesse instante, considere que a lista $\mathcal{L}$ de vértices anteriormente visitados esteja vazia. Após três iterações, a partícula está posicionada no vértice $v_4$ como mostra a segunda linha da ilustração. Note que todo o caminho percorrido foi armazenado na lista. Somente no instante $t + 6$ a partícula poderá retornar pelo mesmo caminho percorrido, pois neste instante o vértice $v_3$ foi removido da lista. Observa-se então que um valor adequado para o parâmetro $\tau$ depende fortemente do grau da rede.

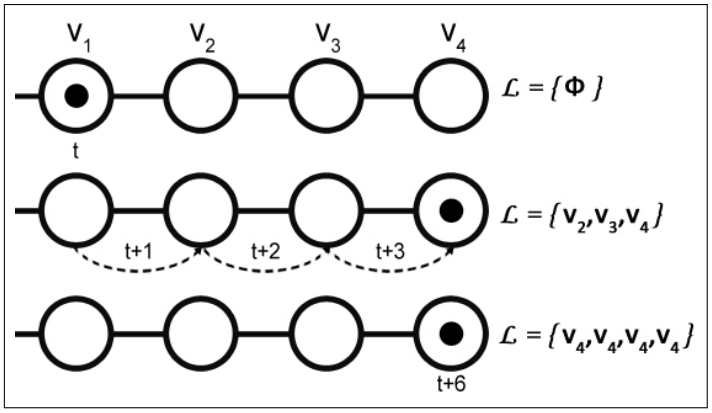



**Figura 4.1:** Ilustração em três estágios do processo de armazenamento dos vértices anteriormente visitados pelas partículas. A primeira linha exibe parte da rede na qual a partícula, representada pela ponto preto, é colocada sobre o vértice $v_1$ no instante $t$. A segunda e terceira linha mostram o movimento da partícula nos instantes intermediários e o posicionamento final respectivamente. Vértices visitados são adicionados na lista $\mathcal{L}$.

Foi proposto um escore de *outlier* baseado na contagem de visitas recebidas por cada vértice. Assume-se que um *outlier* seja um objeto cuja frequência de visitas é substancialmente diferente dos demais objetos que pertencem à mesma classe. Considere um problema de classificação com um conjunto $\mathcal{C}$ de classes. O escore de *outlier* de um vértice $v_i$ no instante $t$ em relação a um time de partículas de índice $m$ é calculado através da equação

$$escore(v_i, m, t) = \frac{\sum_{j=1}^{|\mathcal{V}|} \left( |N_i^{(m)}(t) - N_j^{(m)}(t)| * owner(j, m, t) \right)}{N_i^{(m)}(t)}, \tag{4.2}$$

em que $N_j^{(m)}(t)$ é a quantidade de visitas que o vértice $v_j$ recebeu até o instante $t$ pelas partículas do time de índice $m$ e $|.|$ retorna o valor absoluto de um número. Como mencionado na descrição do modelo original, a função booleana $owner(j, m, t)$ retorna 1 se o vértice $j$ é dominado pelo time de partículas de índice $m$ na iteração $t$, ou 0 caso contrário. Quanto maior o valor do escore, mais provável que o vértice seja um *outlier*. Além disso, é possível verificar facilmente as seguintes propriedades:

1. $escore(v_i, m, t) \geq 0, \forall\ v_i \in \mathcal{V}$.
2. $escore(v_i, m, t) = 0$ se, e somente se, $N_i^{(m)} = N_j^{(m)}$ | $owner(j, m, t) = 1$ e $m = \underset{c \in \mathcal{C}}{argmax}(N_i^{(c)})$

Um detalhe importante é que o valor máximo do escore depende do número de iterações executadas. Entretanto, nenhum critério de parada é rigorosamente seguido. Vale lembrar que os autores do modelo original sugerem um critério de parada que verifica a variação de uma certa medida entre duas iterações consecutivas como, por exemplo, o número de relativo de visitas (Silva e Zhao, 2012). Se a variação for muito pequena, o processo é interrompido. Dependendo do critério de parada adotado, o processo finalizará antes que todos os vértices sejam visitados. Então, o vetor $N$ de cada vértice deve ser inicializado com 1 a fim de evitar a divisão por zero no cálculo do escore. Quanto maior o tempo de execução do algoritmo, os valores dos escores obtidos tendem a ser mais confiáveis. Isso se baseia no fato que existe um intervalo de tempo para que as partículas encontrem a fronteira entre territórios de times diferentes. Uma análise posterior deverá ser realizada sobre vértices não visitados, isto é, aqueles que não pertencem a nenhum time de partículas.

As principais etapas da técnica são descritas no Algoritmo 4. A saída gerada é uma lista contendo o valor do escore de cada vértice em ordem decrescente. Vértices do topo da lista serão os mais prováveis *outliers* da base de dados considerada. Como o algoritmo não diz quais dados são *outliers*, a quantidade deles pode ser especificada por um parâmetro. Dessa forma, basta analisar apenas os dados posicionados no topo da lista de escore.

**Algoritmo 4** Algoritmo de detecção de *outliers* via competição e cooperação de partículas

**Entrada:** Base de dados $\mathcal{X}$, conjunto de rótulos $l$, parâmetros $w_{min}$, $w_{max}$, $\Delta$, $p_{pref}$ e $\tau$.
**Saída:** Lista contendo o valor de escore de *outlier* dos vértices em ordem decrescente.

1: Gerar uma rede conexa $\mathcal{G}$ a partir da base de dados $\mathcal{X}$.
2: Configurar os $l$ rótulos para os objetos considerados como classe normal.
3: Colocar uma partícula em cada vértice rotulado na etapa anterior.
4: Inicializar o vetor $N_i$ da matriz $N$ com $N_i(1) = \mathbb{1}$.
5: Executar o método modificado de competição e cooperação de partículas sobre a rede $\mathcal{G}$.
6: Calcular o escore de outlier usando a equação (4.2).
7: Ordenar decrescentemente a lista de escore de *outlier*.

## 4.2 Simulações sobre bases de dados artificiais

Para fins de ilustração, os experimentos foram executados sobre bases bidimensionais que contêm duas classes apenas. As partículas são colocadas sobre os dados rotulados no início do processo. A rede de cada base de dados não foi ilustrada para facilitar a visualização dos objetos. Como o modelo modificado é utilizado apenas para detecção de *outliers*, não serão apresentados os resultados da classificação dos dados normais. Os parâmetros do modelo original são fixados em $p_{grd} = 0.6$, $\Delta = 0.4$, $w_{min} = 0.05$ e $w_{max} = 1$ para todas as simulações deste Capítulo. Segundo as simulações apresentadas no artigo do modelo original, um bom resultado na classificação de algumas bases pode ser obtido utilizando esses valores.

A primeira simulação é feita sobre a base de dados ilustrada na Figura 4.2 (a) composta por um *cluster* gaussiano e um *cluster* em forma de anel. O objetivo desta simulação é verificar se a técnica proposta consegue detectar *outliers* em bases de dados cujos *clusters* formam estruturas não triviais. Embora não existem *outliers* nesta base de dados, considera-se que os objetos posicionados na borda do *cluster* gaussiano são os mais prováveis de serem considerados *outliers* uma vez que estão em uma região de baixa densidade. A capacidade de identificar estruturas como o *cluster* com formato de anel é muito desejada por algoritmos de agrupamento de dados, pois bases reais comumente apresentam *clusters* não esféricos. Nesta simulação, assume-se que as classes são conhecidas e uma pequena quantia de exemplos rotulados de cada classe esteja disponível. Quando o processo de classificação pelo modelo de competição e cooperação de partículas termina, o escore de *outlier* de cada objeto é calculado e apresentado na Figura 4.2 (b). Quanto maior o valor do escore de um objeto, maior será seu tamanho e sua cor tenderá ao vermelho. Nesta base de dados, os marcadores $p_1$ e $p_2$ indicam dois objetos que a técnica *LOF*, algoritmo baseado em densidade para detecção de *outliers*, não é capaz de identificar. A técnica proposta, pelo contrário, atribuiu um escore significativamente superior se comparado aos objetos localizados no centro do *cluster* gaussiano. As oscilações nos valores do escore para objetos próximos ocorreram devido à rede $kNN$ gerada e à quantidade e posição dos exemplos previamente rotulados que interferem na qualidade do resultado. Como muitos dos objetos vizinhos de $p_1$ e $p_2$ pertencem ao *cluster* em forma de anel, a tarefa de classificação torna-se

complicada. Outro detalhe importante diz respeito à escolha do conjunto de dados previamente rotulados. Embora apenas um conjunto específico de dados previamente rotulados foi considerado na análise, escolhas aleatórias dos rótulos não provocam resultados muito diferentes desde que os rótulos do *cluster* em forma de anel estejam bem distruídos.

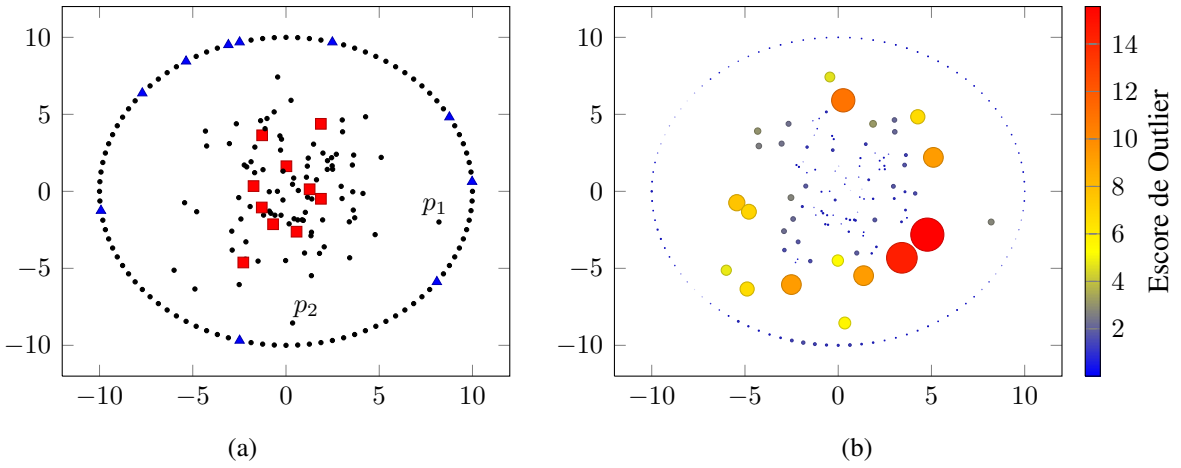



**Figura 4.2:** Base de dados bidimensional formada por um *cluster* em forma de anel e um *cluster* com distribuição gaussiana $\mathcal{N}(0, 2.8)$. (a) Configuração inicial da base de dados na qual os objetos rotulados são representados por triângulos e quadrados. (b) Configuração final da base de dados. As cores e os tamanhos dos objetos indicam o escore de *outlier* de acordo com a medida proposta.

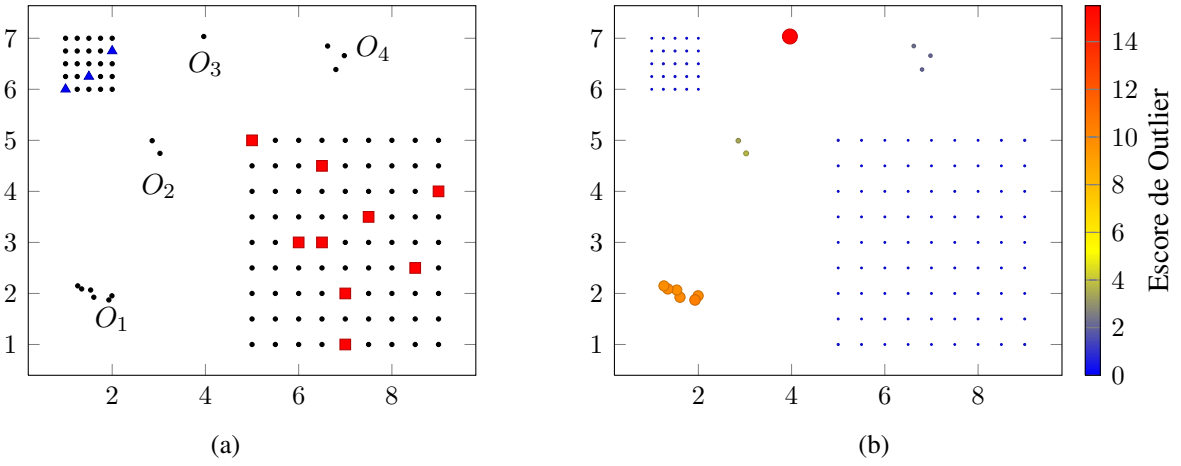



**Figura 4.3:** Base de dados bidimensional formada por dois *clusters* com diferentes tamanhos e densidades. Os *outliers* são denotados pelos marcadores $O_1$, $O_2$, $O_3$ e $O_4$. (a) Configuração inicial da base de dados na qual os objetos rotulados são representados por quadrados e triângulos. (b) Configuração final da base de dados. As cores e os tamanhos dos objetos indicam o escore de *outlier* de acordo com a medida proposta.

A fim de analisar a capacidade da técnica proposta em detectar pequenos grupos de *outliers* quando a base de dados contém *clusters* de tamanhos e densidades diferentes, foi gerada uma base artificial ilustrada na Figura 4.3 (a). Os grupos de *outliers* são indicados pelos marcadores $O_1$, $O_2$, $O_3$ e $O_4$. O total de exemplos rotulados de cada classe é proporcional ao tamanho dos

*clusters*, caso contrário, o time de partículas da menor classe poderá invadir com mais facilidade o maior *cluster*. Boa parte das técnicas de detecção de *outliers* apenas identifica objetos isolados tendo problemas para detectar o grupo *outlier* $O_1$. Outra dificuldade é na identificação do grupo $O_2$, pois os objetos estão localizados entre dois *clusters* de objetos normais. Conforme os escores ilustrados na Figura 4.3 (b), a técnica proposta conseguiu identificar todos os *outliers* dessa base. A utilização de rótulos da classe normal faz com que os *outliers* estejam longe dos vértices casa. Consequentemente, os escores dos *outliers* serão maiores, pois as partículas raramente os visitam. Observou-se também que a técnica proposta tem habilidade para manipular objetos que formam *clusters* de diferentes densidades desde que a quantidade de rótulos de cada classe seja proporcional ao tamanho dos *clusters*. A influência do número de rótulos no processo de classificação pode ser vista no artigo original que descreve a técnica.

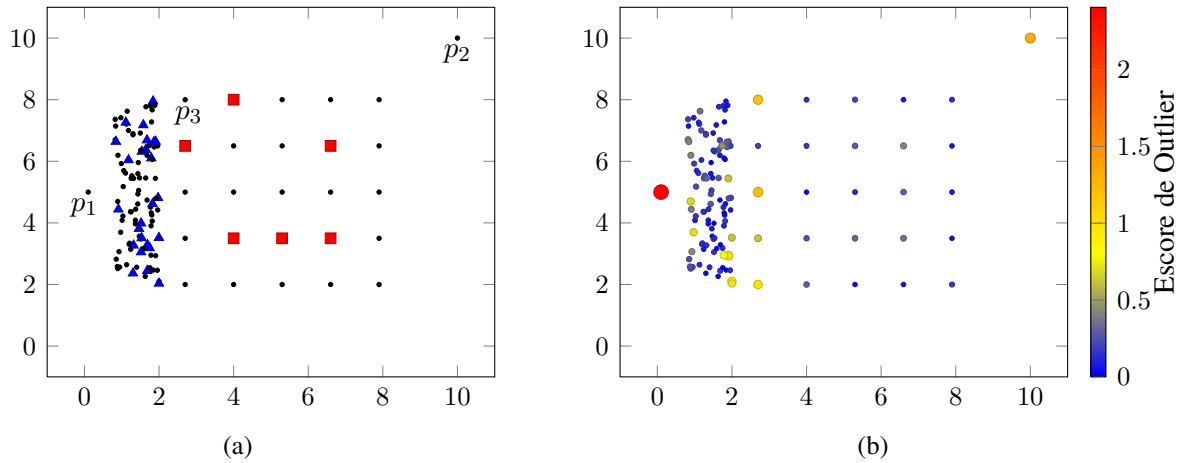



**Figura 4.4:** Base de dados bidimensional formada por um *cluster* denso e alongado de 100 objetos e um *cluster* esparso de 25 objetos. Dois *outliers* $p_1$ e $p_2$ são acrescentados na base. (a) Configuração inicial da base de dados na qual os objetos rotulados são representados por triângulos e quadrados. (b) Configuração final da base de dados. As cores e os tamanhos dos objetos indicam o escore de *outlier* de acordo com a medida proposta.

A última base artificial considerada é ilustrada na Figura 4.4 (a). Nesta base de dados, assume-se a existência de dois *outliers* destacados pelos marcadores $p_1$ e $p_2$. A base contém duas classes sendo uma representada pelo *cluster* alongado e a outra representada pelo *cluster* esparso. Cada ponto $p = (x, y)$ do primeiro *cluster* é tal que $x \in \mathcal{U}(0.8, 2)$ e $y \in \mathcal{U}(2, 8)$ (distribuições uniformes cujos parâmetros são os valores mínimos e máximos permitidos). As técnicas de detecção de *outliers* mais propícias para este caso são aquelas baseadas em densidade, pois conseguem identificar facilmente o objeto $p_1$ como *outlier*. Como a distância de $p_1$ ao grupo mais próximo é menor que a distância entre os objetos da outra classe, duas interpretações frequentemente são feitas por técnicas de outras categorias. Uma delas é considerar apenas o *cluster* alongado como normal e os demais objetos como *outliers*, pois esse *cluster* contém a maioria dos dados e a distância entre seus dados é pequena. Outra interpretação é considerar apenas o objeto $p_2$ como *outlier* visto que o objeto $p_1$ está a uma distância pequena do *cluster* alongado. Essa limitação é devido ao uso somente da distância de cada objeto em relação aos seus vizinhos. A técnica proposta não apresenta essa limitação, pois a rede permite

identificar o padrão formado pelos dados. Outro motivo é que cada objeto é comparado apenas com os demais objetos que pertencem ao mesmo *cluster*. Os resultados ilustrados na Figura 4.4 (b) apontam que a técnica proposta atribuiu um escore de *outlier* maior para os objetos $p_1$ e $p_2$ alcançando uma precisão de $100\%$. Também é possível observar que alguns objetos aparecem destacados na cor amarela. Eles estão, em grande parte, localizados na fronteira entre *clusters*. Nota-se que o *cluster* esparso possui um padrão sugerindo que o objeto $p_3$ ilustrado na Figura 4.4 (a) pertença a este *cluster*. Neste caso, esse objeto deve ser classificado como normal visto que está de acordo com aquele padrão. No entanto, esse mesmo objeto pode ser classificado como *outlier* se ele pertence ao *cluster* alongado já que está em uma região de menor densidade em relação aos objetos do mesmo *cluster*. Essa incerteza na classificação é um dos motivos que levaram a técnica proposta a atribuir um escore relativamente alto para este objeto. Deve-se ressaltar que a identificação desses *clusters* não é uma tarefa trivial em agrupamento de dados.

As simulações anteriores mostraram o funcionamento da técnica proposta. Observou-se que a técnica atingiu bons resultados em bases de dados com diferentes tipos de *clusters*. Vale dizer que a técnica ranqueará os dados de acordo com a medida proposta mesmo se não houver *outliers* na base de dados. Cabe ao especialista decidir em uma análise posterior se os dados ranqueados como *top-outliers* deverão ser rotulados como normais ou não.

## 4.3 Simulações sobre bases de dados reais

As simulações a seguir utilizam duas bases de dados do repositório da UCI[1] as quais foram pré-processadas a fim de conter somente atributos numéricos e instâncias sem atributos com valores ausentes. A classe majoritária foi considerada como classe normal, a classe minoritária foi escolhida como classe *outlier* e as demais classes foram removidas. Amostragem foi necessária para que instâncias da classe *outlier* não superem em número as instâncias da classe normal. Como exemplos da classe *outlier* não possuem rótulos conhecidos, somente um time de partículas foi utilizado. Assim, não há competição e o único propósito do modelo nestes experimentos será detectar *outliers* ao invés de classificar objetos. Em todas as simulações, a precisão da técnica proposta é obtida analisando apenas os vértices com maior grau do escore de *outlier*.

**Breast Cancer (Original):** a base contém $683$ instâncias divididas em duas classes: classe benigna ($444$ instâncias) e classe maligna ($239$ instâncias). Para as simulações, o atributo ID e o atributo classe foram removidos, e foi gerado um subconjunto contendo $454$ instâncias, dez das quais são selecionadas aleatoriamente da classe maligna. A rede construída é a união de uma rede $17NN$ e uma árvore geradora mínima com $45$ dados previamente rotulados ($10\%$) e parâmetro $\tau$ fixado em $3$. Esse processo garante que a rede seja conexa.

**Shuttle:** consiste de $58000$ instâncias divididas em sete classes, porém somente a classe de número $6$ ($11478$ instâncias) e a classe de número $2$ ($13$ instâncias) foram consideradas. Foram selecionadas aleatoriamente $1000$ instâncias da classe $6$ para compor a classe normal e

[1] `http://archive.ics.uci.edu/ml/`

$10$ instâncias da classe $2$ para compor a classe *outlier*. Uma rede $7NN$ é construída para essa base com $100$ dados previamente rotulados (cerca de $10\%$) e parâmetro $\tau = 3$.

A Tabela 4.1 exibe a porcentagem de verdadeiros *outliers* detectados por cada técnica. A precisão é calculada em termos dos objetos com maior valor do escore de *outlier*. Cada resultado é uma média de $20$ execuções. Na base de dados *Wisconsin Breast Cancer*, a técnica conseguiu identificar a maioria das instâncias da classe maligna obtendo precisão de $79.00\%$, isto é, encontrou $8$ de $10$ *outliers* em boa parte das execuções. Este resultado é próximo daqueles obtidos pelas técnicas *LOF* e *SSOD*. No entanto, nenhuma das técnicas alcançou valor máximo da precisão. Estes resultados sugerem que algumas instâncias da classe normal estão localizadas em uma região de baixa densidade ou que as instâncias da classe outlier estão misturadas as instâncias da classe normal. Embora os resultados da técnica *LOF* são os melhores para as duas bases de dados, quando o parametro $k$ tem valor próximo de $200$ e $10$ respectivamente, não há uma maneira eficiente de determinar o melhor valor do parâmetro para cada base. Essa desvantagem também ocorre com a técnica *SSOD* que não apresenta resultados satisfatórios quando instâncias da classe normal não formam cluster esféricos.

Observou-se que a técnica proposta apresentou melhor resultado que outras três técnicas sobre a base de dados Shuttle. Altos valores do desvio padrão são devido à falta de um critério de parada bem definido e por não existir uma heurística para distribuição dos rótulos. O ideal é que os rótulos estejam distribuídos uniformemente de maneira similar àquela realizada para o *cluster* em forma de anel da base artificial ilustrada na Figura 4.2 (a). Vale ressaltar que um número pequeno de iterações resultará em baixa precisão já que vértices podem não receber visitas ou que a diferença entre o número de visitas dos vértices da classe normal e da classe *outlier* seja pequena.

**Tabela 4.1:** Comparação entre a técnica proposta e técnicas tradicionais de detecção de outliers utilizando duas bases de dados do repositório da UCI. Cada célula exibe a precisão e o desvio padrão.

| | **LOF** | **Outrank-b** | **NDoT** | **SSOD** | **Técnica proposta** |
|---|---|---|---|---|---|
| **Breast Cancer** | $81.50 \pm 11.37$ | $0.00 \pm 0.00$ | $80.00 \pm 13.76$ | $81.50 \pm 9.88$ | $79.00 \pm 11.19$ |
| **Shuttle** | $77.50 \pm 7.86$ | $31.50 \pm 6.71$ | $55.00 \pm 11.00$ | $7.50 \pm 4.44$ | $64.50 \pm 17.31$ |

## 4.4 Análise do parâmetro proposto

A fim de compreender o impacto do parâmetro $\tau$ sobre a acurácia da técnica proposta, foi feita uma simulação sobre uma rede $7NN$ gerada a partir da base de dados da Figura 4.3. Variando o valor deste parâmetro no intervalo de $1$ e $7$, observou-se que houve uma queda da precisão para valores pequenos de $\tau$ como pode ser visto na Figura 4.5. A acurácia é medida usando apenas os top-$12$ *outliers* da lista de escore. Essa sensibilidade é completamente reduzida quando o valor é próximo de $6$, quando a precisão atinge o valor máximo de $100\%$. Se

o valor de $\tau$ ultrapassa o grau máximo da rede, a acurácia diminui significativamente. Esse resultado é esperado uma vez que o processo de propagação de rótulos é prejudicado.

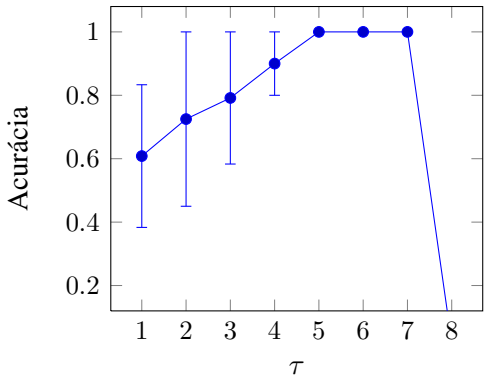



**Figura 4.5:** Acurácia da técnica proposta sobre a base de dados bidimensional da Figura 4.3 para alguns valores do parâmetro $\tau$. Utilizou-se uma rede $7NN$ com o propósito de conectar os *outliers* com os objetos normais. Cada ponto equivale a 20 simulações computacionais e a barra vertical indica o desvio padrão.

Além do grau da rede, o valor do parâmetro $\tau$ deve ser ajustado de acordo com a disponibilidade de rótulos. Muitos rótulos também restringem a caminhada das partículas, pois a quantidade de vértices não ocupados será menor. Dependendo das configurações estabelecidas, uma partícula poderá ficar por um longo período sobre o mesmo vértice. Se um vértice *outlier* é ocupado por uma partícula durante muito tempo, ele poderá ser classificado incorretamente como normal.

## 4.5 Utilidade da informação rotulada

Técnicas de aprendizado semissupervisionado são úteis quando se conhece uma pequena quantidade $m$ de rótulos de uma base com $n$ registros ($m \ll n$). Geralmente, os modelos atingem melhores resultados quanto maior a quantidade de dados rotulados disponíveis para gerá-los, desde que as suposições sobre a distribuição dos dados não rotulados estejam corretas. Nesta seção, deseja-se investigar a eficiência da técnica proposta na detecção de *outliers* em relação à disponibilidade de exemplos rotulados da classe normal, isto é, quando se aumenta o número de partículas que caminham na rede.

Detecção de intrusão é um problema ideal para ser tratado com técnicas semissupervisionadas. Exemplos conhecidos de ataques são escassos e variam muito, ou seja, as formas de ataques mudam constantemente. Abordagens supervisionadas podem falhar nesses casos, pois necessitam que novas categorias de ataques sejam aprendidas. Como rótulos de conexões normais são abundantes, técnicas semissupervisionadas são mais adequadas para esse tipo de problema.

A base de dados *KDD99 Cup*[2] contém registros sobre intrusões em computadores. Cada registro, obtido por meio de simulações de intrusão, representa uma conexão TCP com um IP de origem, um IP de destino e uma sequência de pacotes que trafegaram na rede. As cone-

[2] `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

xões são rotuladas como normais ou como um tipo de ataque específico. Os registros contêm 34 atributos contínuos e 7 atributos categóricos para descrever informações como tipo de protocolo utilizado, duração da conexão, número de operações de criação de arquivo, número de tentativas de login que falharam, entre outros. As categorias de ataques incluem, por exemplo, ataque por negação de serviço (DOS), acesso não autorizado e ações que envolvem a investigação de vulnerabilidades. Utilizou-se essa base no teste a seguir com $1100$ registros escolhidos aleatoriamente dos quais $100$ deles são exemplos de ataques.

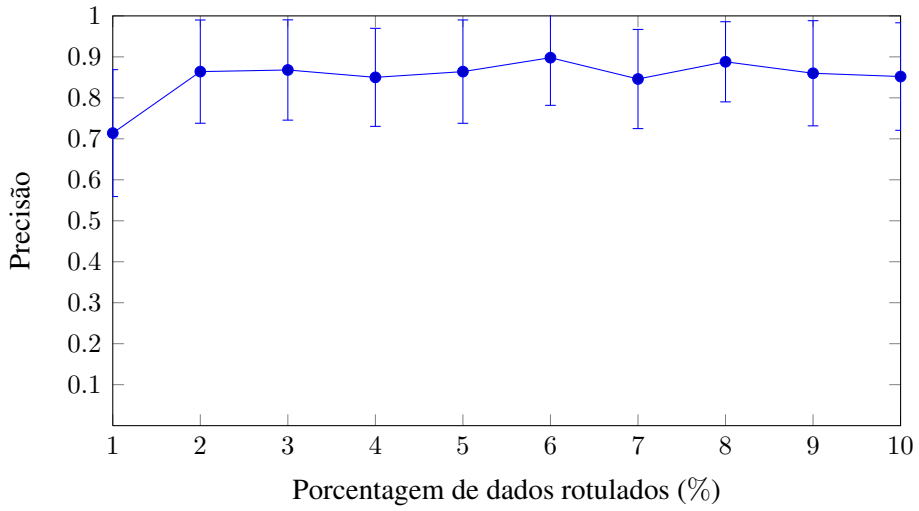



**Figura 4.6:** Precisão da técnica proposta sobre um conjunto de $1100$ exemplos selecionados da base de dados *KDD99 Cup*. Utilizou-se a união de uma árvore geradora mínima (*MST*) e uma rede $13NN$. Cada ponto equivale a 10 execuções com uma quantidade de rótulos fixada e a barra vertical indica o desvio padrão. Parâmetro: $\tau = 3$.

A Figura 4.6 exibe os resultados obtidos pela técnica proposta quando a quantidade de exemplos rotulados é variada no intervalo de $1\%$ a $10\%$ e cada ponto equivale à média de $10$ execuções. O parâmetro $\tau$ foi ajustado para $3$ e a rede gerada é a união de uma árvore geradora mínima (*MST*) e uma rede $13NN$. Nesta simulação, outras redes também foram analisadas, mas os resultados não foram satisfatórios. O motivo de usar uma *MST* é devido a base de dados ter alta dimensão, pois a abordagem $kNN$ não garante que a rede seja conexa. Caso contrário, as subredes precisariam ter pelo menos uma partícula posicionada inicialmente sobre cada uma delas para que os vértices normais sejam visitados. Em algumas execuções a precisão atinge $100\%$ no melhor caso. Como a base de dados é diferente em cada execução, é esperado que exemplos pouco representativos de cada classe sejam selecionados aumentando o desvio padrão.

Realizou-se também um experimento sobre a base de dados da Figura 4.3. Além de variar o número de partículas, verificou-se a eficiência da técnica proposta para alguns valores do parâmetro $\tau$. Neste experimento, ilustrado na Figura 4.7, utilizou-se uma rede $5NN$ sendo a quantidade de rótulos proporcional ao tamanho de cada um dos dois *clusters*. Como pode ser observado, a precisão da técnica aumenta quanto mais dados rotulados são utilizados, exceto quando $\tau = 1$. O desvio padrão, representado por uma barra vertical em cada ponto, não altera significativamente como aumento dos dados rotulados devido ao critério de parada adotado.

Também observou-se que um valor adequado para o parâmetro $\tau$ pode melhorar consideravelmente a precisão da técnica. Quando $\tau = 3$ ou $\tau = 4$ e a porcentagem de dados rotulados está acima de $10\%$, todos os *outliers* foram identificados. A precisão obtida com valores menores do parâmetro $\tau$ não alcança o máximo, mas tende a aumentar com um número maior de dados rotulados. Vale ressaltar que essa base de dados contém um *cluster* pequeno formado por $6$ *outliers*. Valores altos do parâmetro $\tau$ impedem que as partículas permaneçam por um longo período sobre esse *cluster*. Isso explica o porquê dos valores mais altos de $\tau$ produzem melhores resultados.

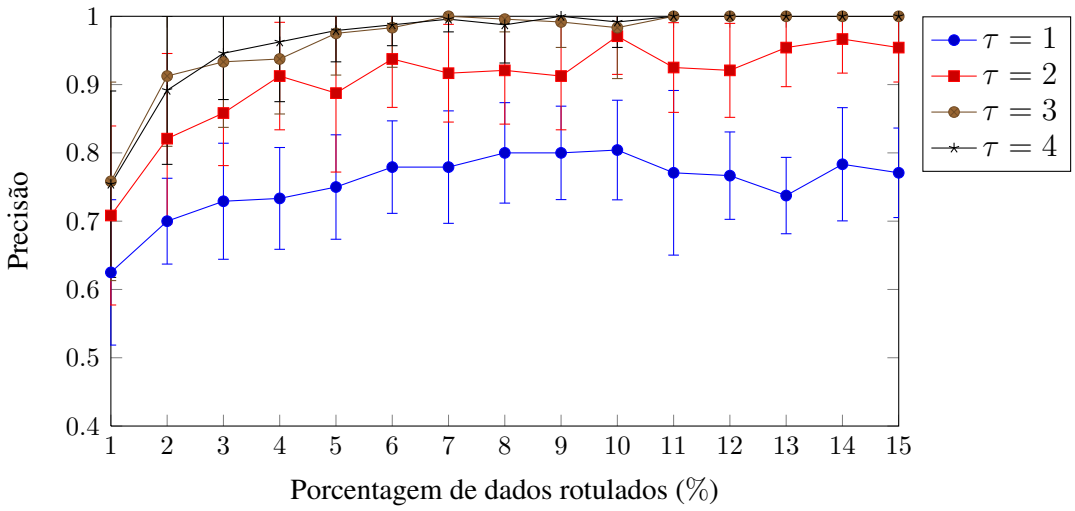



**Figura 4.7:** Simulação realizada sobre uma rede $5NN$ gerada a partir da base de dados da Figura 4.3. A precisão da técnica proposta é verificada para diferentes quantidades de rótulos. Cada ponto é uma média de 20 execuções e a barra vertical indica o desvio padrão.

## 4.6 Efeito da porcentagem de outliers

A suposição que *outliers* são pontos isolados é adotada por várias técnicas como, por exemplo, aquelas baseadas em densidade. Quando esses dados formam pequenos *clusters*, tais técnicas podem falhar uma vez que a densidade local de um *outlier* assemelha-se à densidade local de dados normais. Outro problema ocorre quando eles estão localizados na borda de um *cluster*. Se existem muitos *outliers*, a tendência é incluí-los no *cluster*. Por esses motivos, deve-se avaliar a eficiência das técnicas em relação a quantidade de *outliers*.

A primeira simulação foi realizada sobre a base de dados *Wisconsin breast cancer*, da qual foram removidas as instâncias duplicadas e foram selecionados $212$ objetos da classe benigna. A porção de objetos da classe maligna é variada a fim de simular um problema de classes desbalanceadas. A Figura 4.8 ilustra a precisão da técnica. Cada ponto é uma média de $15$ execuções com uma quantia fixa de objetos selecionados aleatoriamente da classe maligna. Utilizou-se uma rede $13NN$ e $\tau = 4$, pois a escolha desses valores produziram os melhores resultados. Observou-se que a técnica apresentou baixa precisão quando poucos *outliers* são inseridos na base. O alto desvio padrão quando a porcentagem de *outliers* é $1\%$ ($2$ objetos) é esperado uma

vez que algumas instâncias da classe maligna são mais difíceis de identificar. Com essa quantidade fixada, a técnica obteve precisão de $100\%$ em $5$ das $15$ execuções e falhou em apenas uma execução. A medida em que a quantidade aumenta, a eficiência se mantém próxima de $80\%$ mostrando que a técnica é mais adequada quando existe uma quantia significativa de *outliers*. Pode-se dizer que dentro do intervalo considerado, o número de *outliers* não interfere significativamente no resultado. Como mencionado anteriormente, esse aumento pode diminuir a eficiência de algumas técnicas, pois a tendência é que os *outliers* formem grupos. A técnica mostrou-se robusta neste sentido, já que houve baixo desvio padrão.

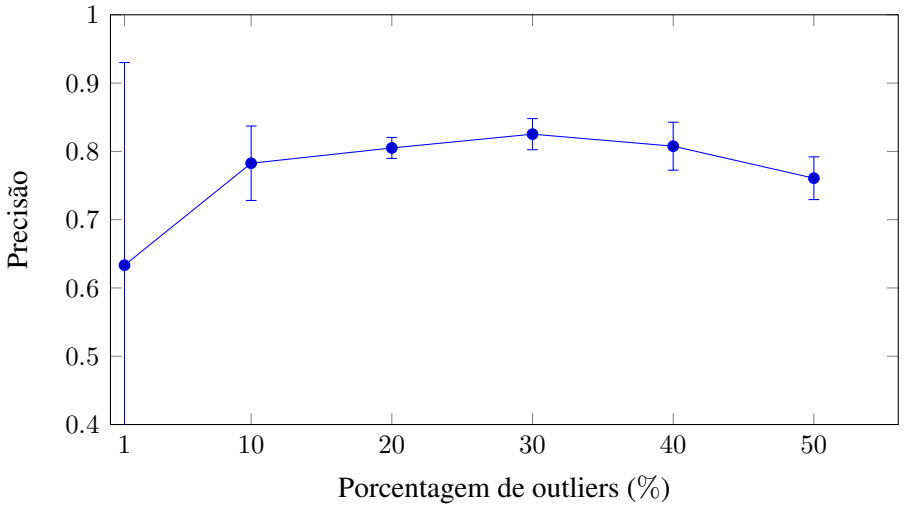



**Figura 4.8:** Efeito da porcentagem de *outliers* na precisão da técnica proposta sobre a base de dados *Wisconsin breast cancer*. Utilizou-se a união de uma árvore geradora mínima e uma rede $13NN$ com o parâmetro $\tau$ fixado em $4$. Cada ponto é uma média de $15$ execuções considerando $212$ exemplos da classe benigna e exemplos da classe maligna como *outliers*.

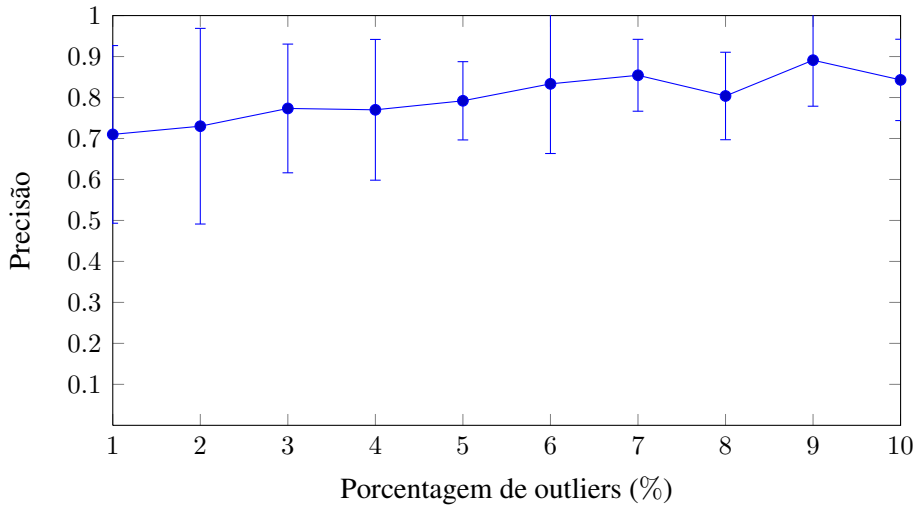



**Figura 4.9:** Efeito da porcentagem de *outliers* na precisão da técnica proposta sobre a base de dados *KDD99 cup*. Utilizou-se a união de uma árvore geradora mínima e uma rede $7NN$ com o parâmetro $\tau$ fixado em $4$. Cada ponto é uma média de $10$ execuções considerando $1000$ exemplos da classe normal e exemplos de ataques como *outliers*.

Também verificou-se a eficiência da técnica proposta sobre a base *KDD99 cup* conforme mostra a Figura 4.9. Os resultados indicam que a técnica apresenta alto desvio padrão quando existem poucos *outliers*. Uma hipótese é que a rede $kNN$ não seja a mais adequada já que a inserção de apenas um *outlier* em uma base representada por uma rede $7NN$ faz com que ele esteja conectado com pelo menos 7 objetos da classe normal. Por outro lado, a rede $kNN$ pode ser adequada quando *outliers* formam grupos, pois diminui a probabilidade de existirem arestas entre objetos de classes diferentes. Análises mais detalhadas sobre a influência da rede $kNN$ serão discutidas na próxima seção.

## 4.7 Influência da rede kNN no resultado obtido

Uma etapa obrigatória das técnicas baseadas em rede é a conversão da base de dados para o formato de rede. Sabe-se que redes pouco representativas afetam diretamente os resultados obtidos por tais técnicas. Eficiência e tolerância a ruídos, por exemplo, podem ser melhoradas com a construção de uma rede esparsa (Jebara et al., 2009). Contudo, a maioria das técnicas semissupervisionadas baseadas em rede não faz um estudo detalhado sobre esse tópico e parte da etapa na qual a rede já está construída. Além disso, construir uma rede apropriada para tarefas como propagação de rótulos requer conhecimento das características da base de dados, do algoritmo utilizado, etc.

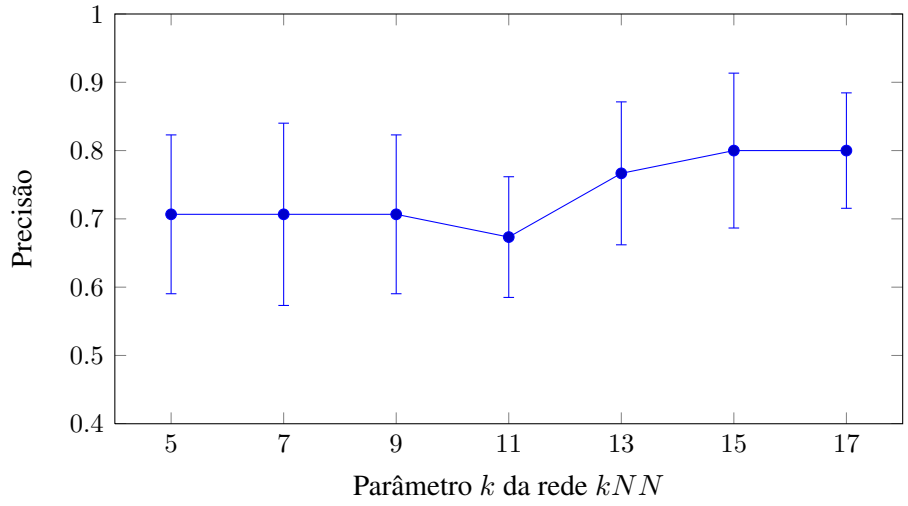
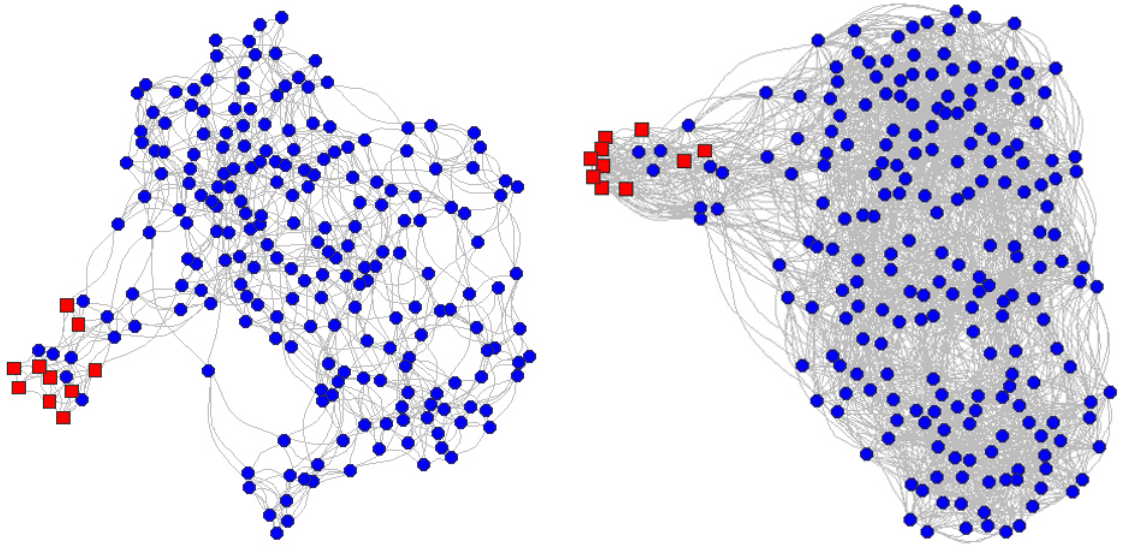



**Figura 4.10:** Precisão da técnica proposta sobre a base de dados *Wisconsin Breast Cancer* para diferentes redes $kNN$. Dentre os 222 objetos considerados, 21 são exemplos previamente rotulados da classe benigna e apenas 10 objetos, escolhidos aleatoriamente, pertencem à classe maligna. Cada ponto equivale a 15 execuções. Parâmetro: $\tau = 4$.

O objetivo desta seção é mostrar a influência da construção da rede $kNN$ nos resultados obtidos pela técnica proposta. Resultados obtidos com outros tipos de redes como a árvore geradora mínima, *mutual* $kNN$ e redes com arestas direcionadas foram inferiores aos obtidos com a rede $kNN$. A primeira simulação é ilustrada na Figura 4.10, a qual foi realizada sobre a base de dados *Wisconsin Breast Cancer*. Foram incluídos apenas 10 exemplos da classe maligna para simular a inserção de um grupo pequeno de *outliers*. Uma rede $kNN$ é gerada em cada

execução e a precisão da técnica é avaliada para diferentes valores de $k$. Quando $k < 10$, a rede resultante poderia ser composta por duas subredes desconexas se as classes dessa base estivessem bem separadas. Entretanto, como desenhado na Figura 4.11 (a), os objetos da classe maligna conectam-se com muitos exemplos da outra classe mesmo para valores baixos de $k$. Essa mistura entre classes comprova o alto desvio padrão e a dificuldade da técnica em detectar todos os *outliers*. Surpreendentemente, os resultados são melhores para valores maiores de $k$. Uma possível interpretação para tais resultados é que o alto grau da rede evita que os caminhos entre exemplos da classe benigna sejam bloqueados devido ao parâmetro $\tau$. Um exemplo da rede com maior grau é ilustrado na Figura 4.11 (b).



(a) (b)

**Figura 4.11:** Representação da base de dados *Wisconsin Breast Cancer* convertida no formato de rede. Os $212$ círculos azuis e $10$ retângulos vermelhos indicam os exemplos da classe benigna e da classe maligna, respectivamente. (a) Rede $5NN$. (b) Rede $17NN$.

A segunda simulação foi realizada sobre a base de dados *KDD99 Cup*. Nota-se, de acordo com a Figura 4.12, a divergência entre os resultados para diferentes redes $kNN$. Valores pequenos de $k$ produziram melhores resultados, pois a inclusão de muitas arestas diminui o isolamento dos *outliers*. Outro detalhe importante é que os *outliers* presentes no conjunto de dados do teste $2$ são mais próximos aos objetos normais e, consequentemente, são mais difíceis de detectar. Entretanto, foi obtido um resultado melhor usando uma rede $5NN$ no teste $2$. Em ambos os testes, essa rede possui componentes desconexos. Porém, no teste $2$, vértices da classe normal ficaram em subredes que continham pelo menos uma partícula. Analisando a quantidade de visitas que cada vértice recebeu, foi constatado que alguns exemplos da classe normal não foram visitados, motivo pelo qual são classificados como *outliers*. Em relação ao teste com a rede $7NN$ cuja precisão alcançada é $89\%$, há alguns *outliers* que não foram detectados, pois receberam bastante visitas. A abordagem $kNN$ não é muito eficiente quando elementos de classes diferentes estão misturados.

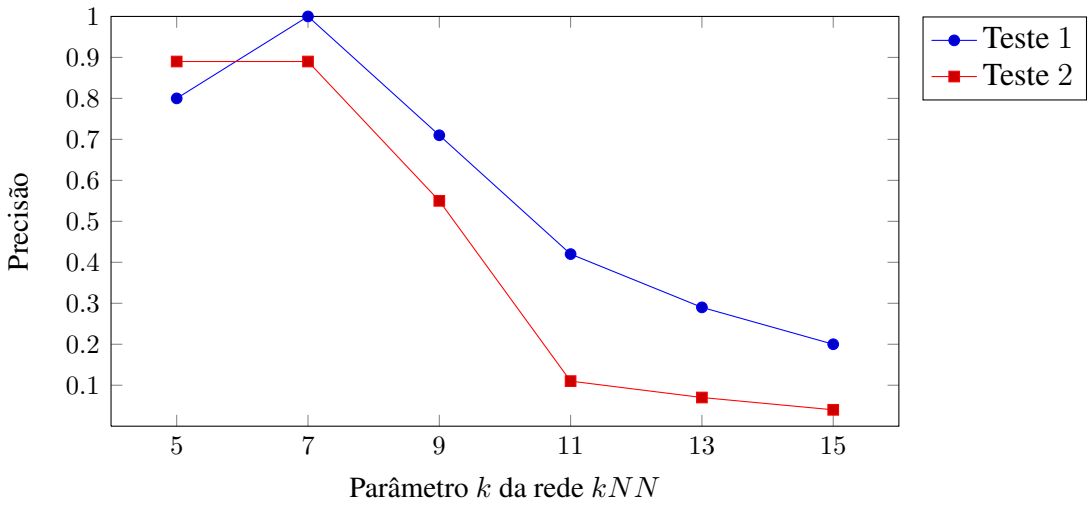



**Figura 4.12:** Precisão da técnica proposta sobre a base de dados *KDD99 Cup* para diferentes redes $kNN$. Cada linha indica os testes realizados sobre um subconjunto dessa base composto por $1000$ exemplos da classe normal e por $100$ exemplos de ataques. Ao todo, são utilizados $100$ rótulos. Parâmetro: $\tau = 4$.

## 4.8 Considerações finais

Neste capítulo, foi apresentada uma técnica baseada em rede para detecção de *outliers*. Para tanto, foi proposta uma medida baseada no mecanismo de competição e cooperação de partículas para caracterização de vértices da rede, o qual foi originalmente proposto para classificação de dados. A medida considera a frequência de visitas recebidas pelos vértices assumindo que *outliers* são aqueles que apresentam um número de visitas muito diferente em relação aos demais vértices de mesma classe.

Após a descrição da técnica, foram analisados os resultados obtidos por meio de simulações feitas em bases artificiais e reais. As bases artificiais foram ilustradas para melhor compreensão da técnica e com apenas dois *clusters* para fins de simplificação. Assim, foi possível constatar que a técnica proposta consegue atribuir um alto escore de *outlier* aos objetos que estão em regiões de baixa densidade. Também verificou-se que a técnica é capaz de manipular bases com *clusters* de formas arbitrárias. Na parte de simulações em bases reais, foram selecionadas bases que fazem parte do repositório da UCI analisando a precisão da técnica de acordo com os *outliers* previamente definidos. Boa parte dos resultados é similar aqueles obtidos pelas técnicas tradicionais de detecção de *outliers*.

Um novo parâmetro foi acrescentado ao modelo, responsável por definir o tamanho do vetor que armazena a lista de vértices visitados por cada partícula. Dessa maneira, a atribuição de um valor adequado para esse parâmetro impede que as partículas realizem ciclos pequenos. Durante a análise deste parâmetro, verificou-se que valores muito altos ou muito baixos não produziram os melhores resultados, sugerindo que um valor ótimo depende da rede construída. A técnica também foi avaliada em relação a proporção de exemplos rotulados e a proporção de *outliers*. Utilizou-se uma base de dados de detecção de intrusão, a qual é frequentemente empregada para avaliar técnicas de detecção de *outliers*.

A grande limitação da técnica proposta está na dificuldade em determinar os melhores valores dos parâmetros para um certo problema. O artigo original que descreve o modelo de competição e cooperação de partículas contém experimentos que auxiliam na determinação de boa parte desses parâmetros. Neste trabalho, a maioria dos parâmetros do modelo original foi fixada a fim de simplificar a análise da técnica proposta. No que se refere ao parâmetro $\tau$, as simulações apresentadas nas seções anteriores indicam que a escolha do valor deste parâmetro depende da rede construída. Em redes $kNN$, um valor adequado seria aquele mais próximo ao valor de $k$.

Apresentar-se-ão as conclusões sobre os resultados obtidos e as ideias para trabalhos futuros no próximo capítulo.

Capítulo 5

# Conclusão

O presente trabalho propõe e analisa uma técnica para detecção de *outliers*. O tópico sobre *outliers* tem sido bastante explorado, pois envolve, por exemplo, a pesquisa sobre novas formas de extração de conhecimento de grandes bases de dados e redução de prejuízo em aplicações financeiras. Contudo, a tarefa não é trivial uma vez que esses dados exibem um comportamento não esperado. Uma das maneiras de identificar esse comportamento é por meio da comparação com o padrão formado por dados normais. Construir uma rede a partir dos dados é uma das estratégias usualmente empregadas para identificar estruturas complexas, pois as redes podem representar mais adequadamente os relacionamentos entre os dados. Isso motivou a utilização de técnicas baseadas em redes para detecção de *outliers*.

A revisão bibliográfica abordou dois assuntos: aprendizado de máquina e detecção de *outliers*. O primeiro assunto diz respeito aos métodos capazes de aprender com a experiência adquirida. Dentre os três paradigmas de aprendizado de máquina apresentados, foi dado ênfase ao paradigma semissupervisionado cuja técnica proposta pertence. Na tentativa de construir bons modelos de classificação de dados, esse paradigma leva em consideração o padrão formado pelos dados não rotulados e uma pequena quantidade de dados rotulados para a construção de um classificador. Isso faz com que tenha grande valor prático na indústria visto que dados não rotulados são abundantes e podem ajudar na classificação. O segundo assunto abordou os principais conceitos e técnicas tradicionais de detecção de *outliers*. Embora seja um tópico de intensa pesquisa, há poucas técnicas semissupervisionadas, as quais são mais adequadas quando poucos exemplos da classe normal são conhecidos.

O Capítulo 4 apresentou a técnica proposta para detecção de *outliers* fundamentada no mecanismo de competição e cooperação de partículas. Foi proposto um escore de *outlier* que considera a frequência de visitas dos vértices da rede. Admitiu-se que vértices pouco visitados pertencem a regiões afastadas enquanto vértices muito visitados pertencem a regiões densas. A

fim de avaliar a eficiência da técnica, foram feitas simulações tanto em bases de dados artificiais quanto em bases de dados reais incluindo a comparação com algoritmos bem conhecidos na literatura. Foi proposto um parâmetro para identificar *clusters outliers* e sua influência na qualidade dos resultados foi apresentada. Outras análises relevantes também foram realizadas como a utilidade da informação rotulada, efeito da porcentagem de *outliers* e influência da rede.

## 5.1 Principais conclusões

Na literatura, a detecção de *outliers* geralmente é tratada com técnicas estatísticas, assumindo que a distribuição dos dados é conhecida, ou com técnicas baseadas em densidade. Recentemente, novas abordagens têm sido propostas, dentre elas, aquelas baseadas em rede. Neste trabalho, foi investigada uma técnica baseada em rede para tratar desse problema. Como esperado, a representação dos dados na forma de rede permite identificar os padrões formados pelos dados auxiliando no processo de detecção de *outliers*.

Foi observado que o mecanismo de competição e cooperação de partículas pode fornecer informações valiosas para caracterização de vértices. Especificamente, foi proposto um escore de *outlier* que faz uso do número de visitas recebidas por cada vértice. Tal abordagem é uma maneira não convencional de identificar *outliers* uma vez que a distância e a densidade dos objetos não são calculadas explicitamente.

Com base nos resultados apresentados, foi possível verificar que a técnica consegue manipular bases de dados com diferentes tipos de *clusters* e com diferentes quantidades de *outliers*. A técnica é robusta para bases de dados que contenham pequenos grupos de *outliers* os quais estão distantes das partículas e consequentemente recebem poucas visitas. Também foram feitas simulações envolvendo *clusters*, de tamanhos e densidades diferentes, formados por objetos da classe normal. Nestas situações, as técnicas baseadas apenas no cálculo da distância entre objetos costumam falhar, pois não levam em consideração a densidade ao redor de cada objeto. Pelo contrário, a técnica proposta atingiu bons resultados e conseguiu ser superior, em determinados casos, à técnica baseada em densidade *LOF*.

Como a técnica proposta segue a abordagem semissupervisionada, foi necessário verificar a precisão em relação à quantidade de rótulos. Houve uma pequena melhora na precisão com o aumento de rótulos e uma pequena redução no desvio padrão. Além disso, os melhores resultados foram obtidos quando os exemplos rotulados estão bem espalhados sobre a base de dados. Bases de dados que apresentam estruturas mais simples geralmente não necessitam mais que $10\%$ de dados rotulados para obter alta precisão. Por outro lado, quando classes possuem estruturas mais complexas como o *cluster* em forma de anel, é ideal que exemplos previamente rotulados sejam bem distribuídos. No que se refere à porcentagem de *outliers*, foi constatado que houve melhores resultados quando existe uma quantia significativa de *outliers*, pois a técnica é adequada para identificar pequenos grupos de *outliers* conforme mencionado anteriormente. Quando há poucos *outliers* na base, eles são conectados com muitos exemplos

da classe normal devido ao método de formação da rede. Essa formação afeta diretamente a precisão e, por isso, é mandatório que a rede represente adequadamente a base de dados.

Os principais resultados apresentados neste trabalho originaram a seguinte publicação:

Fabio Zamoner, Liang Zhao (2013) *A network-based semi-supervised outlier detection technique using particle competition and cooperation*. *Proceedings of the Brazilian Conference on Intelligent Systems*, 2013 (pp. 225-230).

## 5.2 Trabalhos futuros

Diversas questões ainda permanecem abertas como uma análise sobre o impacto do posicionamento inicial das partículas no resultado final do algoritmo, quantidade mínima de exemplos rotulados para um desempenho satisfatório e estudo de técnicas para geração da rede. A construção da rede é uma das etapas importantes de técnicas baseadas em redes, pois os resultados dependem da qualidade da rede gerada. Uma rede que não represente adequadamente a relação entre os dados pode afetar consideravelmente a precisão na detecção de *outliers*. Logo, o estudo sobre outras formas de construção da rede faz-se necessário.

Pode-se obter diversas informações derivadas do processo de competição e cooperação de partículas como: distância da partícula em relação ao seu vértice casa, número de vezes que um vértice trocou de dono, potencial dos vértices, atribuição de um tempo de vida para cada partícula, etc. Apenas o número de visitas foi considerado na composição do escore de *outlier*. Então, outras medidas poderão ser empregadas para caracterização dos vértices da rede sob diferentes aspectos. Em relação à atribuição de um tempo de vida para cada partícula, um critério de parada é implicitamente definido visto que o término do processo ocorre quando não houver partículas caminhando na rede.

Uma das aplicações mais conhecidas no contexto de *outlier* é a detecção de regiões de destaque visual em imagens. Qualquer método que manipula matriz assimétrica pode ser usado nesse contexto. No caso do modelo de partículas, a imagem pode ser considerada como uma rede cujos vértices representam os pixels da imagem. Com a escolha de uma medida de similaridade adequada, a hipótese é que regiões salientes apresentem alguma característica diferente das demais regiões. Detecção de *outliers* em imagens é um problema de caráter prático.

Outra possibilidade é propor uma nova definição para um objeto fora do padrão nomeado *outlier* estrutural. Tal definição será fundamentada na conformidade de componentes da rede em relação aos demais por meio da análise de medidas de rede previamente selecionadas. Essas medidas podem ser globais se estiverem relacionadas a caminhos ou circuitos, intermediárias se relacionadas a grupo de vértices e locais se relacionadas a um único vértice ou aresta. O *outlier* estrutural não precisará estar fisicamente distante dos demais objetos, mas afastado em relação as características extraídas das medidas de rede. Consequentemente, deverá ser investigada a função ou característica diferenciada que tais *outliers* possuem em bases de dados originalmente na forma de redes.

# Referências Bibliográficas


Agyemang, M., Barker, K. e Alhajj, R. (2006). A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis*, *10*, 521–538.

Alpaydin, E. (2010). *Introduction to machine learning*. The MIT Press.

Barnett, V. e Lewis, T. (1995). *Outliers in statistical data* (3rd ed.). Other Wiley Editorial Offices.

Berton, L., Huertas, J., Araújo, B. e Zhao, L. (2010). Identifying singular nodes in complex networks by using random walks measure. In *Proceddings of the IEEE Congress on Evolutionary Computation (IEEE CEC)* (Vol. 1, pp. 2891–2896).

Breitenbach, M. e Grudic, G. Z. (2005). Clustering through ranking on manifolds. In *In proceedings of the 22nd international conference on machine learning* (pp. 73–80). ACM Press.

Breunig, M., Kriegel, H.-P., Ng, R. T. e Sander, J. (2000). LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM Sigmod International Conference on Management of Data* (pp. 93–104). ACM.

Breve, F., Zhao, L., Quiles, M., Pedrycz, W. e Liu, J. (2011). Particle competition and cooperation in networks for semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, *99*(PrePrints).

Chandola, V., Banerjee, A. e Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58.

Chapelle, O., LinkSchölkopf, B. e LinkZien, A. (2006). *Semi-supervised learning* (1st ed.). The MIT Press.

Chaudhary, A., Szalay, A. S., Szalay, E. S. e Moore, A. W. (2002). Very fast outlier detection in large multidimensional data sets. In *DMKD'02*. ACM Press.

Chen, Z., Hendrix, W. e Samatova, N. (2011). Community-based anomaly detection in evolutionary networks. *Journal of Intelligent Information Systems*, 1-27.

Clauset, A. (2005). Finding local community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, *72*(2), 026132.

Cook, D. J. e Holder, L. B. (2006). *Mining graph data*. John Wiley & Sons.

Costa, L. d. F., Rodrigues, F. A., Hilgetag, C. C. e Kaiser, M. (2009). Beyond the average: Detecting global singular nodes from local features in complex networks. *EPL (Europhysics Letters)*, *87*(1), 18008.

Davies, L. e Gather, U. (1993). The identification of multiple outliers. *Journal of the American Statistical Association*, *88*(423), 782–792.

Eberle, W. e Holder, L. (2007). Discovering structural anomalies in graph-based data. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops* (pp. 393–398). Washington, DC, USA: IEEE Computer Society.

Ester, M., Kriegel, H.-p., Jörg, S. e Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In (pp. 226–231). AAAI Press.

Faceli, K., Lorena, A. C., Gama, J. e Carvalho, A. C. P. L. F. de. (2011). *Inteligência artificial: Uma abordagem de aprendizado de máquina*. LTC.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*(3-5), 75 - 174.

Gan, G., Ma, C. e Wu, J. (2007). *Data clustering theory, algorithms, and applications*. SIAM, Society for Industrial and Applied Mathematics.

Gao, J., Cheng, H. e Tan, P.-N. (2006). Semi-supervised outlier detection. In *Proceedings of the 2006 acm symposium on applied computing* (pp. 635–636). New York, NY, USA: ACM.

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

Halkidi, M., Batistakis, Y. e Vazirgiannis, M. (2002a). Clustering validity checking methods: part ii. *ACM SIGMOD Record*, *31*(3), 19–27.

Halkidi, M., Batistakis, Y. e Vazirgiannis, M. (2002b). Cluster validity methods: part I. *ACM SIGMOD Record*, *31*, 40–45.

Hautamaki, V., Karkkainen, I. e Franti, P. (2004). Outlier detection using k-nearest neighbour graph. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)* (Vol. 3, pp. 430–433). Washington, DC, USA: IEEE Computer Society.

Hawkins, D. M. (1980). *Identification of outliers* (1st ed.). Chapman and Hall, London.

Hawkins, S., He, H., Williams, G. J. e Baxter, R. A. (2002). Outlier detection using replicator neural networks. In *Proceedings of the 4th international conference on data warehousing and knowledge discovery* (pp. 170–180). London, UK: Springer-Verlag.

He, Z., Huang, J., Xu, X. e Deng, S. (2004). Mining class outliers: Concepts, algorithms and applications. In Q. Li, G. Wang e L. Feng (Eds.), *Advances in web-age information management* (Vol. 3129, p. 589-599). Springer Berlin Heidelberg.

He, Z., Xu, X. e Deng, S. (2003). Discovering cluster based local outliers. *Pattern Recognition Letters*, *2003*, 9–10.

Hodge, V. e Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85–126.

Hu, T. e Sung, S. Y. (2003). Detecting pattern-based outliers. In *Pattern recognition letters 24* (p. 3059-3068).

Huang, Z. e Zeng, D. D. (2006, oct.). A link prediction approach to anomalous email detection. In *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on* (Vol. 2, p. 1131 -1136).

Hubballi, N., Patra, B. K. e Nandi, S. (2011). NDoT: Nearest neighbor distance based outlier detection technique. In S. O. Kuznetsov, D. P. Mandal, M. K. Kundu e S. K. Pal (Eds.), *PReMI* (Vol. 6744, p. 36-42). Springer.

Jain, A. K. e Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Jain, A. K., Murty, M. N. e Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, *31*(3), 264–323.

Jebara, T., Wang, J. e Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 441–448). New York, NY, USA: ACM.

Jiang, D., Tang, C. e Zhang, A. (2004). Cluster analysis for gene expression data: a survey. *IEEE Transactions on Knowledge and Data Engineering*, *16*(11), 1370–1386.

Jin, W., Tung, A. K. H. e Han, J. (2001). Mining top-n local outliers in large databases. In *Proceedings of the seventh acm sigkdd international conference on knowledge discovery and data mining* (pp. 293–298). New York, NY, USA: ACM.

Karypis, G., Han, E.-H. e Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75.

Knorr, E. M. e Ng, R. T. (1997). A Unified Approach for Mining Outliers. In *Proceedings of the 1997 Conference of the Centre for Advanced Studies on Collaborative Research* (pp. 219–222). IBM Press.

Kou, Y. e Lu, C.-t. (2006). Spatial weighted outlier detection. In *Proceedings of SIAM Conference on Data Mining* (pp. 614–618). SIAM.

Kou, Y., Lu, C.-T., Sirwongwattana, S. e Huang, Y.-P. (2004). Survey of fraud detection techniques. In *Networking, Sensing and Control, 2004 IEEE International Conference on* (Vol. 2, pp. 749–754).

Li, Y., Fang, B., Guo, L. e Chen, Y. (2007). Network anomaly detection based on TCM-KNN algorithm. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security* (pp. 13–19). New York, NY, USA: ACM.

Macqueen, J. B. (1967). Some methods of classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Mitchell, T. M. (1997). *Machine learning*. Mc Graw-Hill.

Moonesignhe, H. e Tan, P.-N. (2006). Outlier detection using random walks. *Tools with Artificial Intelligence, IEEE International Conference on*, *0*, 532-539.

Morettin, P. A. e Bussab, W. d. O. (2003). *Estatística básica* (5th ed.). Saraiva.

Newman, M. E. J. (2004). Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, *38*(2), 321-330.

Newman, M. E. J. e Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*(2), 026113.

Noble, C. C. e Cook, D. J. (2003). Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 631–636). New York, NY, USA: ACM.

Oliveira, J. V. de e Pedrycz, W. (2007). *Advances in fuzzy clustering and its applications*. Wiley.

Pan, F. e Wang, W. (2006). Anomaly detection based-on the regularity of normal behaviors. In *Systems and Control in Aerospace and Astronautics, 2006. ISSCAA 2006. 1st International Symposium on* (p. 6 pp.-1046).

Portnoy, L., Eskin, E. e Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)* (pp. 5–8). New York, NY 10027: Department of Computer Science, Columbia University.

Pothen, A., Simon, H. D. e Liou, K.-P. (1990). Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal Matrix Analysis Applications*, *11*(3), 430–452.

Quan, Q., Hong-Yi, C. e Rui, Z. (2009). Entropy based method for network anomaly detection. In *Dependable Computing, 2009. PRDC '09. 15th IEEE Pacific Rim International Symposium on* (p. 189-191).

Quiles, M., Liang, Z., Alonso, R. L. e Romero, R. A. F. (2008). Particle competition for complex network community detection. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *18*(3), 033107.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Ramaswamy, S., Rastogi, R. e Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *SIGMOD Record*, *29*, 427–438.

Rattigan, M. J. e Jensen, D. (2005). The case for anomalous link detection. In *Proceedings of the 4th international workshop on multi-relational mining* (pp. 69–74). New York, NY, USA: ACM.

Reichardt, J. e Bornholdt, S. (2004). Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. *Physical Review Letters*, *93*(21), 218701.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, *1*(1), 27 - 64.

Sheikholeslami, G., Chatterjee, S. e Zhang, A. (1998). WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th vldb conference* (pp. 428–439).

Shekhar, S., Lu, C.-T. e Zhang, P. (2001). Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 371–376). New York, NY, USA: ACM.

Shetty, J. e Adibi, J. (2005). Discovering important nodes through graph entropy the case of Enron email database. In *Proceedings of the 3rd international workshop on Link discovery*

(pp. 74–81). New York, NY, USA: ACM.

Silva, T. e Zhao, L. (2012). Network-based stochastic semisupervised learning. *Neural Networks and Learning Systems, IEEE Transactions on*, *23*(3), 451-466.

Singh, S. e Markou, M. (2004). An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering*, 396–407.

Strang, G. (2009). *Introduction to linear algebra* (4th ed.). Wellesley - Cambridge Press.

Su, X. e Tsai, C.-L. (2011). Outlier detection. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, *1*(3), 261-268.

Sun, H., Bao, Y., Zhao, F., Yu, G. e Wang, D. (2004). CD-Trees: An efficient index structure for outlier detection. In Q. Li, G. Wang e L. Feng (Eds.), *Advances in web-age information management* (Vol. 3129, p. 600-609). Springer Berlin / Heidelberg.

Sun, J., Qu, H., Chakrabarti, D. e Faloutsos, C. (2005). Neighborhood formation and anomaly detection in bipartite graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining* (pp. 418–425). Washington, DC, USA: IEEE Computer Society.

Sun, P., Chawla, S. e Arunasalam, B. (2006). Mining for outliers in sequential databases. In *In siam international conference on data mining*. SIAM.

Szummer, M. e Jaakkola, T. (2002). Partially labeled classification with Markov random walks. In *Advances in neural information processing systems* (pp. 945–952). MIT Press.

Tan, P.-N., Steinbach, M. e Kumar, V. (2005). *Introduction to data mining* (1st ed.). Addison Wesley.

Theodoridis, S. e Koutroumbas, K. (2008). *Pattern recognition* (4th ed.). Academic Press.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience.

Xu, R. e Wunsch, D. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, *16*(3), 645–678.

Yu, D., Sheikholeslami, G. e Zhang, A. (2002). FindOut: Finding outliers in very large datasets. *Knowledge and Information Systems*, *V4*(4), 387–412.

Zhang, T., Ramakrishnan, R. e Livny, M. (1997). BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, *1*, 141–182.

Zhou, H. (2003). Distance, dissimilarity index, and network community structure. *Physical Review E*, *67*(6), 061901.

Zhu, X. (2005). *Semi-supervised learning with graphs*. Unpublished doctoral dissertation, Pittsburgh, PA, USA. (AAI3179046)

Zhu, X. e Goldberg, A. B. (2009). *Introduction to semi-supervised learning*. Morgan & Claypool.