UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS DA NATUREZA
DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA
DA COMPUTAÇÃO

Rotulação Automática de Grupos com Aprendizagem de Máquina Supervisionada

Lucas Araújo Lopes

Dissertação de Mestrado

Teresina
26 de março de 2014

Lucas Araújo Lopes

Rotulação Automática de Grupos com Aprendizagem de Máquina Supervisionada

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal do Piauí, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Aprendizagem de Máquina.

Orientador: Vinicius Ponte Machado

Co-orientador: Ricardo de Andrade Lira Rabêlo

Teresina
2014

Lopes, Lucas Araújo.
Rotulação Automática de Grupos com Aprendizagem de Máquina Supervisionada
59 páginas
Dissertação (Mestrado) - Universidade Federal do Piauí. Departamento de Computação.

1. Aprendizagem de Máquina
2. Agrupamento
3. Rotulação

I. Universidade Federal do Piauí. Programa de Pós-graduação em Ciência da Computação. Departamento de Computação.

Comissão Julgadora:

Prof. Dr.
Vinicius Ponte Machado - UFPI
(Orientador)

Prof. Dr.
Ricardo de A. Lira Rabêlo - UFPI
(Co-orientador)

Prof. Dr.
André Macêdo Santana - UFPI

Prof. Dr.
Ivan Nunes Silva - USP

"Seja a mudança
que você quer ver no mundo."

– Dalai Lama

Agradecimentos

Agradeço em primeiro lugar, a Deus, pela vida, família, amigos, dons e oportunidades.

Agradeço a meus pais, Raimundo e Fernanda, por todo o carinho, atenção, amor, confiança, ensino e inspiração em toda a minha vida.

Agradeço ao meu orientador, Vinicius Ponte Machado, pela paciência, confiança, estima, inspiração, incentivo, amizade e por me aceitar como orientando. Foi um grande prazer tê-lo como tutor durante essa jornada. Meu profundo agradecimento!

Agradeço ao meu co-orientador, Ricardo de Andrade Lira Rabêlo, por desde a faculdade ter me apoiado e incentivado. Pela honra de termos trabalhados juntos mais uma vez e por ter me proporcionado grandes oportunidades que geraram grandes conquistas. Meu eterno agradecimento!

Agradeço a todo o PPgCC, especialmente aos professores André Soares, André Macêdo, Raimundo, Pedro, Kelson, por toda atenção e apoio.

Agradeço ao meu irmão, Iago, à minha irmã, Pâmella, e ao meu cunhando Márcio, por todo o amor, atenção e paciência.

Agradeço ao meu avôs Helder (*in memoriam*) e Jesus (*in memoriam*; e às minhas avós Bernadete e Sulamita (*in memoriam*), por toda a fé.

Agradeço a cada um de meus tios, tias, primos e primas, pelos grandes momentos juntos.

Agradeço aos mais que amigos, irmãos do San Diego, Lucas (II), Vítor, Wilton Jr., Jairon, Luís, Lorena, Larissa, João Neto, Maurício, Érica, Iago (Pingo), Marconni, Giovanni, Débora, por tudo que temos vivido.

Agradeço a cada um de meus professores, de colégio e faculdade, pelos conhecimentos adquiridos, em especial aos professores Francisco Araújo (Chiquim), José Ferreira, Ricardo Sekeff, Ricardo Queiroz e Harilton Araújo, pela confiança e ensinamentos além do curso.

Agradeço às professoras Rosianni e Amélia e aos professores de matemática Anderson, Laércio e Geraldo.

Agradeço aos amigos da turma de mestrado Romuere, Flávio, Jonathas, Ronaldy, Manoel, Igo, Nathan, Hidelbrando, Bruno e Vilmar.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro fornecido durante parte dessa jornada.

Agradeço a todos que contribuíram direta ou indiretamente com a realização deste trabalho.

Resumo

O problema de agrupamento (*clustering*) tem sido considerado como um dos problemas mais relevantes dentre aqueles existentes na área de pesquisa de aprendizagem não-supervisionada (subárea de Aprendizagem de Máquina). Embora o desenvolvimento e aprimoramento de algoritmos que solucionam esse problema tenha sido o principal foco de muitos pesquisadores o objetivo inicial se manteve obscuro: a compreensão dos grupos formados. Tão importante quanto a identificação dos grupos (*clusters*) é sua compreensão e definição. Uma boa definição de um *cluster* representa um entendimento significativo e pode ajudar o especialista ao estudar ou interpretar dados. Frente ao problema de compreender *clusters* – isto é, de encontrar uma definição ou em outras palavras, um *rótulo* – este trabalho apresenta uma definição para esse problema, denominado *problema de rotulação*, além de uma solução baseada em técnicas com aprendizagem supervisionada, não-supervisionada e um modelo de discretização. Dessa forma, o problema é tratado desde sua concepção: o agrupamento de dados. Para isso, um método com aprendizagem não-supervisionada é aplicado ao problema de *clustering* e então um algoritmo com aprendizagem supervisionada irá detectar quais atributos são relevantes para definir um dado *cluster*. Adicionalmente, algumas estratégias são utilizadas para formar uma metodologia que apresenta em sua totalidade um rótulo (baseado em atributos e valores) para cada grupo fornecido. Finalmente, essa metodologia é aplicada em quatro bases de dados distintas.

Palavras-chave: Aprendizagem de Máquina, Agrupamento, Rotulação

Abstract

The clustering problem has been considered as one of the most important problems among those existing in the research area of unsupervised learning (a subarea of Machine Learning). Although the development and improvement of algorithms that deal with this problem has been focused by many researchers, the main goal remains obscure: the understanding of generated clusters. As important as identify clusters is understand and define them. A good definition of a cluster means a relevant understanding and can help the specialist to study or interpret data. Facing the problem of comprehend clusters – in other words, create labels – this paper presents a definition for this problem (labeling problem) and also a solution involving techniques based on supervised and unsupervised learning and a discretization model as well. Thus, this problem is dealt similar to a real problem, being initialized from grouping data. For this, an unsupervised learning technique is applied to the clustering problem and then a supervised learning algorithm will detect which are the relevant attributes in order to define a given cluster. Additionally, some strategies are used to create a methodology that presents a label (based on attributes and their values) for each cluster provided. Finally, this methodology is applied in four distinct databases.

Keywords: Machine Learning, Clustering, Labeling

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução à Dissertação

Neste capítulo é discutido a importância do tema abordado. Além disso, apresenta-se o principal objetivo, a estrutura do trabalho e as contribuições científicas obtidas durante a etapa de desenvolvimento e pesquisa.

1.1 Introdução

O problema de agrupamento (*clustering*) tem sido considerado como um dos problemas mais relevantes dentre aqueles existentes na área de pesquisa de aprendizagem não-supervisionada (subárea de Aprendizagem de Máquina).

Clustering se refere ao processo de particionar um conjunto de dados (ou objetos) em subconjuntos menores denominados *clusters* ou simplesmente grupos. Nesse processo, os objetos que possuem similaridades em suas características tendem pertencer a um mesmo *cluster* enquanto que objetos com características diferentes tendem pertencer a *clusters* distintos.

Conforme Han et al. (2011), o processo de *clustering* tem sido amplamente utilizado em diversas aplicações como inteligência empresarial, reconhecimento de padrões em imagem, busca na *Web*, biologia, segurança, entre várias outras. No contexto em-

presarial, o agrupamento pode ser utilizado para organizar uma grande quantidade de clientes em grupos. Isso facilita o desenvolvimento de estratégias de negócio para uma melhor gestão de relacionamento com o cliente. Na área de reconhecimento de padrões em imagem pode ser utilizado, por exemplo, para aumentar a precisão de sistemas de reconhecimento de escrita. Em buscas na *Web*, *clustering* pode organizar a informação em assuntos similares apresentado-a de maneira concisa.

Ainda conforme Han et al. (2011), independente da área de aplicação, o problema de *clustering* tem sido bastante estudado e explorado. Sendo um problema difícil, os algoritmos desenvolvidos para essa tarefa enfrentam alguns problemas, como:

- escalabilidade: em muitas áreas, as bases de dados possuem uma grande quantidade de objetos chegando a milhões ou até mesmo bilhões;
- representação do conhecimento: a habilidade de lidar com diferentes tipos de atributos;
- não-linearidade: embora alguns algoritmos sejam limitados a gerar *clusters* com forma convexa, muitos problemas possuem objetos não linearmente separáveis exigindo grupos com diferentes formas;
- domínio do problema: algumas técnicas exigem parâmetros de entrada para seu funcionamento, como por exemplo, a quantidade *clusters* a serem gerados;
- ruídos: em muitas aplicações as bases de dados podem conter objetos com valores desconhecidos, não existentes ou até mesmo com erro;
- novos objetos: alguns algoritmos são capazes de reorganizar seus *clusters* formados à medida que novos objetos são apresentados; outros precisam recalcular todo o processo;
- dados com alta dimensão: alguns problemas descrevem seus objetos em uma

grande quantidade de características e as técnicas de agrupamento devem estar preparadas para lidar com isso;

- restrições: em alguns casos existe a necessidade de obedecer determinadas condições de restrição;

- *interpretação e usabilidade*: os especialistas precisam que os resultados apresentados sejam interpretáveis, compreensíveis e úteis. Muitas vezes, os resultados precisam ser relacionados à interpretação do próprio especialista, não sendo claro quais características foram importante durante o processo.

Assim, embora o desenvolvimento e aprimoramento de algoritmos que enfrentem esses problemas tenha sido foco de muitos pesquisadores, poucos trabalhos são relacionados à interpretação dos *clusters* formados. Segundo Tzerpos (2001), muitos pesquisadores se preocuparam com os demais problemas e não têm demonstrado atenção necessária ao problema específico de melhor compreendê-los.

A compreensão se deve, principalmente, aos valores apresentados pelas características mais importantes de seus objetos. Assim, este conjunto de valores relevantes representam uma definição para um *cluster* qualquer – isto é, um *rótulo* – capaz de fornecer ao especialista um melhor entendimento sobre o mesmo. A interpretação de um rótulo pode, por sua vez, implicar em diversas soluções ou otimização do problema abordado.

Voltando ao contexto de inteligência empresarial, um especialista de uma empresa qualquer ao observar os rótulos fornecidos pode facilmente identificar quais são as principais características que definem os diferentes grupos de clientes. Uma vez conhecida as características, a empresa pode elaborar modelos de negócio ou estratégias específicas para um determinado grupo.

Em outro contexto, uma universidade deseja identificar os diferentes grupos de alunos e aplicar ações corretivas que possam elevar o desempenho de seus estudantes. Com

um algoritmo de agrupamento os *clusters* são fácilmente formados. Entretanto, existe a necessidade de descobrir o que caracteriza cada *cluster* formado. A existência de um rótulo permite a identificação de quais características definem um grupo. Assim, o rótulo pode ser útil para a identificação de quais características necessitam de ações corretivas e até mesmo o quão intensa ela deve ser baseada nos valores das características. Dessa forma, a compreensão de *clusters* por intermédio de rótulos pode contribuir de diversos modos com a elaboração da solução ou otimização de um problema.

1.2 Objetivo

O objetivo dessa dissertação consiste em apresentar uma abordagem capaz de rotular *clusters* a fim de esclarecer, orientar e ajudar um especialista. Os rótulos gerados devem ser capazes de identificar as principais características – bem como seus conjuntos de valores – responsáveis pela definição de um determinado *cluster*.

1.3 Estrutura Organizacional

Após a introdução, o Capítulo 2 apresenta as técnicas utilizadas na abordagem proposta: *K-means*, *Redes Neurais Artificiais* e métodos de *discretização*. No Capítulo 3, são apresentados o problema e a abordagem desenvolvida para a rotulação de *clusters*. O Capítulo 4 mostra os resultados obtidos da aplicação do modelo criado em 4 bases de dados distintas. Finalmente, o Capítulo 5 discute alguns problemas encontrados e apresenta sugestões de trabalhos futuros.

1.4 Contribuições Científicas

No decorrer do desenvolvimento deste trabalho foram publicados em congressos nacionais artigos relacionados à abordagem proposta com os resultados preliminares obtidos.

As publicações foram:

- LOPES, L. A.; MACHADO, V. P.; RABÊLO, R. de A. L. *Automatic Labeling of Groupings through Supervised Machine Learning. X Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, 2013, Fortaleza.

- LOPES, L. A.; MACHADO, V. P.; RABÊLO, R. de A. L. *Clusters Labeling Through Multi-Layer Perceptron Algorithm. XI Simpósio Brasileiro de Automação Inteligente (SBAI)*, 2013, Fortaleza.

Adicionalmente, o seguinte trabalho foi aceito para publicação (internacional):

- LOPES, L. A.; MACHADO, V. P.; RABÊLO, R. de A. L. *Automatic Cluster Labeling through Artificial Neural Networks. IEEE, World Congress on Computational Intelligence (WCCI)* em *International Joint Conference on Neural Networks (IJCNN)*, 2014, Pequim.

Capítulo 2

Referencial Teórico

Neste capítulo é apresentado o embasamento teórico para auxiliar a compreensão do presente trabalho. Na primeira seção apresenta-se uma introdução à área de Aprendizagem de Máquina (Machine Learning), mais especificamente em seus dois principais paradigmas de aprendizagem: supervisionada e não-supervisionada, com foco em duas técnicas utilizadas no modelo proposto no Capítulo 3. Em seguida, é dada uma explanação sobre modelos de discretização usados neste trabalho.

2.1 Aprendizagem de Máquina

Conforme Mitchell (1997), a área de Aprendizagem de Máquina (AM) lida com o estudo de métodos computacionais que permitem programas de computadores obter uma melhoria, de forma autônoma, em uma determinada tarefa por meio de experiências. Diferente das metodologias computacionais tradicionalmente utilizadas, o uso de AM lida com o problema de modo que a própria máquina irá encontrar, após um processo de aprendizagem, uma hipótese que melhor o define. Para isso, a AM baseia-se em ideias de um conjunto diversificado de disciplinas incluindo Inteligência Artificial (IA), probabilidade e estatística, complexidade computacional, teoria da informação, psico-

logia, neurobiologia, teoria de controle e filosofia, tendo aplicações nas mais diversas áreas do conhecimento.

Visualizando a área de AM de uma forma bem ampla, podemos resumi-la basicamente em dois paradigmas de aprendizagem: supervisionada e não-supervisionada. Na primeira, busca-se a criação de um modelo preciso em relação à predição de valores para novos dados enquanto que na segunda o objetivo é encontrar características que podem resumir os dados. Em ambos os casos existe uma busca por um modelo capaz de generalizar dados desconhecidos sendo diferenciados basicamente pela existência de um rótulo[1] (resposta) presente nos dados utilizados na aprendizagem supervisionada. Adicionalmente, existe uma outra abordagem – não utilizada neste trabalho – conhecida como aprendizado semi-supervisionado na qual existe uma tentativa de aprimorar um classificador criado a partir de dados rotulados com o uso de amostras não-rotuladas (Barber, 2012).

2.1.1 Aprendizagem Supervisionada

Os métodos de aprendizagem supervisionada são capazes de modelar padrões existentes nos dados. Um elemento pertencente a uma base de dados qualquer pode ser definido por um conjunto de pares de valores que contêm uma característica (atributo) e seu respectivo valor. A quantidade de características definem a dimensão do problema abordado enquanto que seus respectivos valores servirão de entrada (*input*) para um dado modelo, sendo essas utilizadas diretamente em seu processo de aprendizagem.

Na aprendizagem supervisionada, para cada conjunto de valores de entrada existe um respectivo conjunto de respostas (*output*) que deverão ser apresentadas ao processo. Pode-se imaginar um professor (ou um supervisor) que indica a saída que deverá ser apresentada para determinados valores de entrada. Dessa forma, esse processo pode ser

[1] Os termos "dados rotulados" e "dados não-rotulados" referem-se a um conjunto de dados que apresentam ou não, respectivamente, uma saída desejada para uma determinada combinação dos valores de entrada. Não confundir o termo com o rótulo gerado para a nomeação de *clusters* apresentados como resultado deste trabalho.

modelado, entre outras coisas, como um problema de classificação onde cada amostra, contendo um conjunto de características como entrada e um conjunto de respostas como saída, é utilizada durante as etapas de treinamento e teste.

Os métodos utilizados nas etapas de treinamento e teste podem variar e devem ser ajustados conforme a técnica aplicada. Nesse contexto, a etapa de treinamento consiste, basicamente, em utilizar parte dos dados para a elaboração de um modelo classificador. Essa parte dos dados é conhecida como conjunto de treinamento. Os valores de entrada são atribuídos ao modelo e seu resultado é então comparado à resposta esperada. O modelo produzido é reajustado sempre que houver erro ou até que se atinja uma margem de erro aceitável. Quando esta condição for finalmente satisfeita – ou que se atinja uma quantidade máxima de iterações – segue-se para a etapa de testes na qual a parte restante dos dados (denominada conjunto de teste) será utilizada para medir a acurácia do modelo produzido. As amostras presentes no conjunto de treinamento e conjunto de teste são mutuamente exclusivas.

Formalmente, Barber (2012) apresenta uma definição para a aprendizagem supervisionada: dado um conjunto de dados $D = \{(x_n, y_n), n = 1, ..., N\}$ contendo N elementos, o objetivo é criar um modelo capaz de aprender a relação existente entre os valores de entrada x e os valores de saída y de modo que quando uma nova entrada x^* é fornecida a predição de sua saída y^* é precisa.

Redes Neurais Artificiais

Conforme Silva et al. (2010), as Redes Neurais Artificiais (RNAs) são modelos computacionais que possuem capacidade de aquisição e manutenção de conhecimento baseado em informações, representados por um conjunto de unidades de processamento. Com inspiração no sistema nervoso de seres vivos, podem ser vistas como uma organização de quatro estruturas menores compostas por:

- neurônios artificiais, que são as unidades de processamento;

- sinapses artificiais (comumente chamadas de pesos sinápticos), representando a grande quantidade de interconexões presentes na rede e armazenando o conhecimento do modelo;
- valores de entrada (*input*);
- e valores de saída (*output*).

Assim, as RNAs possuem diversas características interessantes que as tornam relevante a serem aplicadas no presente trabalho: adaptação por experiência, tolerância a falhas, organização dos dados, armazenamento distribuído, facilidade de prototipagem, habilidade de generalização e, especialmente, capacidade de aprendizado.

A capacidade de extrair o relacionamento existente entre as (diversas) variáveis existentes no problema, via método de treinamento será explorada no presente trabalho em relação a detecção de atributos relevantes ao problema, conforme detalhado no Capítulo 3 além de ser bastante explorada e comprovada em diversos trabalhos (Rodrigues et al., 2008; Battiti, 1994; Setiono e Liu, 1997).

Historicamente, o primeiro neurônio artificial foi publicado por McCulloch & Pitts em 1943 (McCulloch e Pitts, 1943). Os autores propuseram o primeiro modelo matemático inspirado em um neurônio biológico resultando na concepção do que seria o primeiro neurônio artificial. Em seguida, uma rede neural artificial em sua forma mais simples (denominada *Perceptron*) foi apresentada por Rosenblatt em 1958 (Rosenblatt, 1958). A Figura 2.1 ilustra uma rede *Perceptron*.

Cada x_i representa um valor referente às variáveis de entrada e, juntos, formam a camada de entrada da rede. Os pesos sinápticos são representados por w_i. Adicionalmente, existe uma unidade denominada *bias*, cujo valor de entrada (x_0) é uma constante diferente de 0, associada ao neurônio artificial. Durante o processo de treinamento a rede recebe os valores de entrada de uma amostra k ($x_1^{(k)}, ..., x_n^{(k)}$). Tais valores são multiplicados por seus respectivos pesos sinápticos ($w_1, ..., w_n$) e então somados –

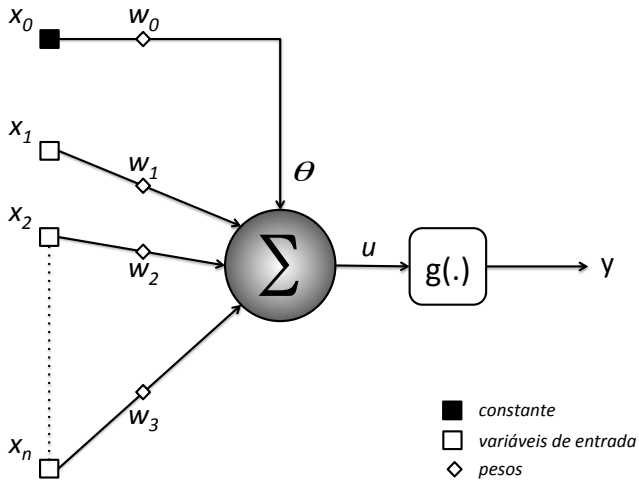

Figura 2.1: Modelo de uma rede *Perceptron* (adaptado de Silva et al. (2010)).

juntamente com o *bias* ($\theta = x_0 \cdot w_0$) – no neurônio artificial (2.1).

$$u = \sum_{i=1}^{n} x_i \cdot w_i + \theta \tag{2.1}$$

O resultado da soma, u, é aplicado a uma função (denominada função de ativação, $g(.)$) que irá apresentar um valor de saída y (2.2). A função de ativação deve ser uma função diferenciável ou parcialmente diferenciável. A função degrau, por exemplo, representa um exemplo de função parcialmente diferenciável que pode ser utilizada.

$$y = g(u) \tag{2.2}$$

Este valor apresentado pela rede é finalmente comparado com o valor desejado da amostra, $d^{(k)}$, avaliando-se uma possível existência de erro (2.3).

$$\epsilon = d^{(k)} - y \tag{2.3}$$

Caso o erro não esteja dentro de um limite aceitável, definido previamente, aplica-se um método de treinamento que irá ajustar os pesos da rede. A equação (2.4) exemplifica um tipo de aprendizado – regra de aprendizado de Hebb, onde η indica a taxa de aprendizagem que controla a precisão da ação corretiva. Esse processo, iterativo, de reajuste se repete até que o erro não exista ou esteja dentro de um limite aceitável – ou ainda que o número máximo de iterações seja extrapolado.

$$w_i^{atual} = w_i^{anterior} + \eta \cdot \epsilon \cdot x^{(k)} \tag{2.4}$$

Após o processo de treinamento ocorre a etapa de testes que tem como objetivo avaliar a performance do modelo gerado. Nesta etapa são apresentados dados desconhecidos pela rede – isto é, dados não apresentados durante a etapa de treinamento – que passam pelas mesmas operações da da etapa anterior. Comparando o resultado apresentado pela rede com o resultado desejado estima-se o quão preciso é o modelo gerado.

Embora a rede *Perceptron* seja restrita à resolução de problemas linearmente separáveis seu modo de funcionamento atraiu diversos pesquisadores desencadeando uma série de trabalhos e pesquisas, conduzindo ao surgimento de diversos novos modelos de redes neurais como *Adaline* (Widrow e Hoff, 1960), redes recorrentes de Hopfield (Hopfield, 1982), redes auto-organizáveis de Kohonen (Kohonen, 1982), entre outras apresentadas em Haykin (2001).

Dentre os modelos de redes neurais artificiais propostos existe a rede *Perceptron* de múltiplas camadas (PMC), ou em inglês *Multilayer Perceptron* (*MLP*). Conforme Silva et al. (2010), as redes PMC se caracterizam pela presença de pelo menos uma camada intermediária (ou camada oculta) de neurônios, localizada entre a camada de entrada e a camada neural de saída da rede. Assim como o *Perceptron*, as redes PMCs possuem uma topologia do tipo *feedforward*, isto é, as saídas dos neurônios de uma dada camada

servem como entrada exclusivamente para neurônios de camadas adiantes. A Figura 2.2 ilustra uma rede *Perceptron* de múltiplas camadas.

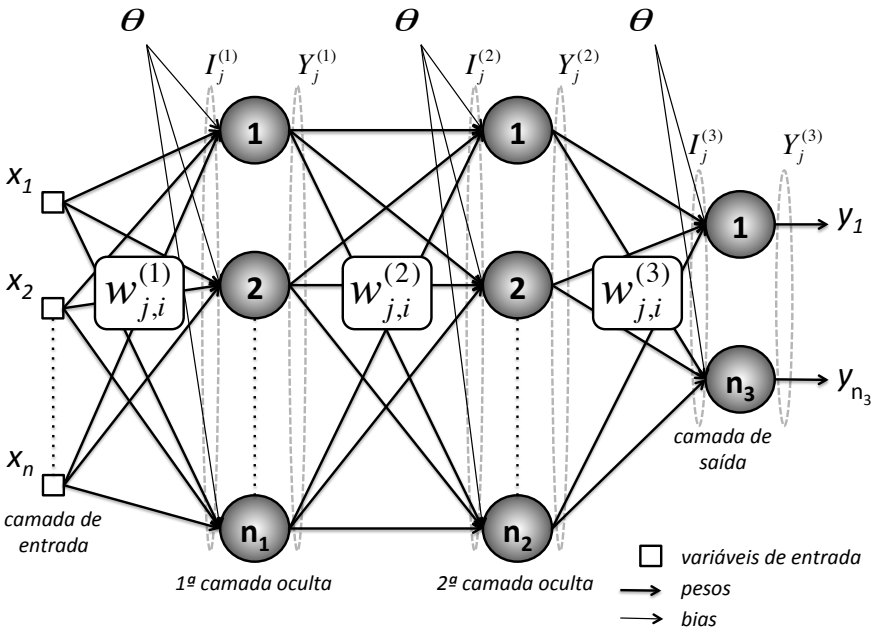

Figura 2.2: Modelo de uma rede *Perceptron* de múltiplas camadas (adaptado de Silva et al. (2010)).

Assim como na rede *Perceptron*, os sinais de entrada de uma rede PMC $(x_1, x_2, ..., x_n)$ possuem pesos associados a cada neurônio da camada seguinte (primeira camada oculta), agora representados por $w_{j,i}^{(1)}$, onde j representa o neurônio e i a entrada. O valor de entrada I de cada neurônio j pertencente à camada 1 é calculado pelo somatório dos produtos dos valores de entrada da camada anterior (2.5).

$$I_j^{(1)} = \sum_{i=0}^{n} x_i \cdot w_{j,i}^{(1)} \tag{2.5}$$

Em seguida, o valor de entrada de um neurônio é aplicado a uma função de ativação $g(.)$. Geralmente, a função mais utilizada é a função tangente hiperbólica embora qualquer outra função totalmente diferenciável possa ser utilizada como por exemplo as funções logística, linear ou gaussiana. Em um neurônio j, para um dado valor de

entrada $I_j^{(1)}$ a função de ativação apresenta, então, um valor de saída $Y_j^{(1)}$ (2.6).

$$Y_j^{(1)} = g(I_j^{(1)}) \tag{2.6}$$

Estes valores de saída multiplicados por seus respectivos pesos $w_{j,i}^{(2)}$ e somados com as unidades de *bias* servirão como entrada para os neurônios da camada seguinte dando origem às entradas $I_j^{(2)}$ da segunda camada oculta. O processo se repete de forma análoga para a segunda camada e, finalmente, a camada de saída apresenta em cada neurônio os valores finais da rede.

De um modo geral, em uma rede PMC qualquer podem existir L camadas ocultas de modo que o processo de propagação dos sinais de entrada ocorre de maneira análoga ao explicado anteriormente. Dessa forma, cada uma das L camadas possuem pesos, representados por $w_{j,i}^{(l)}$ com $1 \leqslant l \leqslant L$, que combinados com os valores de entrada e as unidades de *bias* geram os valores de entrada $I_j^{(l)}$ e, finalmente, os valores de saída $Y_j^{(l)}$.

Cada saída da rede é então comparada com sua respectiva saída desejada. A existência de um erro inaceitável requer a aplicação de algum método de treinamento que terá como objetivo reajustar os pesos da rede. Os métodos mais comumente utilizados com o algoritmo de *backpropagation* é o método do gradiente descendente embora outros métodos, como por exemplo *Levenberg-Marquardt* (Levenberg, 1944; Marquardt, 1963), possam ser utilizados.

Finalmente, após o processo de treinamento ocorre a etapa de teste da rede. Conforme Kohavi (1995), existem 3 principais métodos para validar a capacidade de generalização de um modelo: *holdout*, validação cruzada (*cross-validation*) e *bootstrap*. No método *holdout* – utilizando-se um conjunto exclusivo para testes – é feita uma comparação entre os resultados apresentados pelo modelo gerado e os resultados reais esperados, presente nos elementos utilizados, totalizando na acurácia do modelo avaliado. Em uma variação deste método conhecida como subamostragem aleatória (*random*

subsampling), o processo é repetido M vezes, onde a acurácia final do modelo gerado é dada pela média das acurácias em cada execução.

2.1.2 Aprendizagem Não-Supervisionada

Os métodos de aprendizagem não-supervisionada buscam detectar padrões existentes nos dados a fim de representa-los de forma resumida. Diferente da aprendizagem supervisionada não existe uma resposta para cada conjunto de valores de entrada do problema. Assim, o problema consiste em agrupar diversas amostras em classes distintas de modo que o grau de similaridade entre elementos de um mesmo grupo seja o máximo possível e ainda que o grau de dissimilaridade entre elementos de grupos distintos também seja o máximo possível.

Formalmente, Barber (2012) define aprendizagem não-supervisionada: dado um conjunto de dados $D = \{(x_n), n = 1, ..., N\}$ contendo N elementos, o objetivo é encontrar uma descrição plausível e compacta dos dados. Portanto, aprendizagem não-supervisionada pode ser vista em sua essência como um sinônimo para *clustering* onde pretende-se atribuir classes aos elementos de uma base de dados.

K-means

O agrupamento por *K-means* é um método de agrupamento particional, inicialmente apresentado por MacQueen (1967). Sua função é particionar uma base de dados em K grupos mutualmente exclusivos e indicar a qual grupo cada elemento pertence. Diferentemente de métodos hierárquicos – que trabalham com medidas de dissimilaridade – o *K-means* funciona com base em valores que estimam um grau de similaridade.

Segundo Han et al. (2011), dada uma base de dados, D, contendo N elementos em um espaço Euclidiano, os métodos particionais reorganizam os objetos de D em K grupos (*clusters*), $C_1, ..., C_K$, de modo que $C_i \subset D$ e $C_i \cap C_j = \emptyset$ para $1 \leqslant i, j \leqslant K$ e $i \neq j$, restrito a $K \leqslant N$. Assim, um elemento está associado exclusivamente a um

único grupo.

Conforme Manning et al. (2009), o objetivo do *K-means* é minimizar a média quadrática da distância Euclidiana entre os centros de cada *cluster* e seus respectivos elementos, onde cada centro ou centróide $\vec{\mu}$ de um dado *cluster* c_i é definido como a média de seus elementos, representados individualmente por $\vec{x}$ (2.7).

$$\vec{\mu}(c_i) = \frac{1}{|c_i|} \cdot \sum_{\vec{x} \in c_i} \vec{x} \tag{2.7}$$

A princípio, os K centróides são gerados – normalmente de forma aleatória embora estratégias acerca de suas inicializações possam ser aplicadas. Em seguida, cada elemento $\vec{x}$ é atribuído ao *cluster* que possui o centróide mais próximo. Em uma iterações posterior, os centróides são recalculados conforme (2.7) e os elementos são novamente reatribuídos aos centróides mais próximos. Esse processo se repete até um critério de parada ou que o algoritmo tenha convergido – isto é, que os centróides não sofram alterações – totalizando na definição dos grupos.

Por fim, uma avaliação de quão bem os centróides são capazes de representar seus *clusters* pode ser realizada pelo método dos mínimos quadrados (*residual sum of squares* - *RSS*), expresso pela soma do quadrado das diferenças de cada elemento ao seu centróide, em todos os *clusters* (2.8) e (2.9).

$$RSS_{c_i} = \sum_{\vec{x} \in c_i} |\vec{x} - \vec{\mu}(c_i)|^2 \tag{2.8}$$

$$RSS = \sum_{i=1}^{K} RSS_{c_i} \tag{2.9}$$

O funcionamento do *K-means* tem complexidade computacional $O(t \cdot K \cdot N)$, na qual t é o número de iterações, K é o número de *clusters* a serem gerados e N é o número de amostras, de modo que geralmente tem-se $t, K \ll N$. O algoritmo 1 apresenta

seu funcionamento, embora um estudo mais aprofundado deva ser feito em relação ao método de inicialização dos centróides e ao realizar a normalização dos dados antes de executar o algoritmo.

Algoritmo 1: *K-means*

$(\vec{\mu}_1, ..., \vec{\mu}_K) \leftarrow inicializarCentroides(\{\vec{x}_1, ..., \vec{x}_N\}, K)$;
while condição de parada não for satisfeita **do**
  for $i \leftarrow 1 : K$ **do**
    $c_i \leftarrow \{\}$;
    for $n \leftarrow 1 : N$ **do**
      $j \leftarrow arg\ min_{j'} \left|\vec{\mu}_{j'} - \vec{x}_n\right|$;
      $c_j \leftarrow c_j \cup \{\vec{x}_n\}$; (*reorganizar os elementos*)
    end for
    $\vec{\mu}_i \leftarrow \frac{1}{|c_i|} \cdot \sum_{\vec{x} \in c_i} \vec{x}$; (*recalcular os centróides*)
  end for
end while
return $\{\vec{\mu}_1, ..., \vec{\mu}_K\}$;

Finalmente, a Figura 2.3 demonstra o comportamento do *K-means*. Na Figura 2.3a, podemos ver diversos elementos de uma base de dados representados em duas dimensões e a inicialização (aleatória) de dois centróides. Portanto, o objetivo consta de dividir os elementos em 2 grupos distintos. A Figura 2.3b mostra a atribuição de cada elemento ao centróide mais próximo. Em seguida, os centróides são recalculados conforme os *clusters* inicialmente definidos na primeira iteração (Figura 2.3c). O processo de reatribuição dos elementos aos *clusters* bem como o recálculo dos centróides se repetem até que o algoritmo converge na iteração 9 (Figura 2.3d). Por fim, o deslocamento dos centróides ao longo das iterações ocorridas é apresentado na Figura 2.3e.

Devido à sua simplicidade e eficiência, o *K-means* foi escolhido para lidar com o problema de agrupamento (*clustering*) encontrado neste trabalho. Além disso, seu estudo é bastante difundido na literatura estando presente em diversos trabalhos (Kanungo et al., 2002; Oyelade et al., 2010; Liu e Yu, 2009).

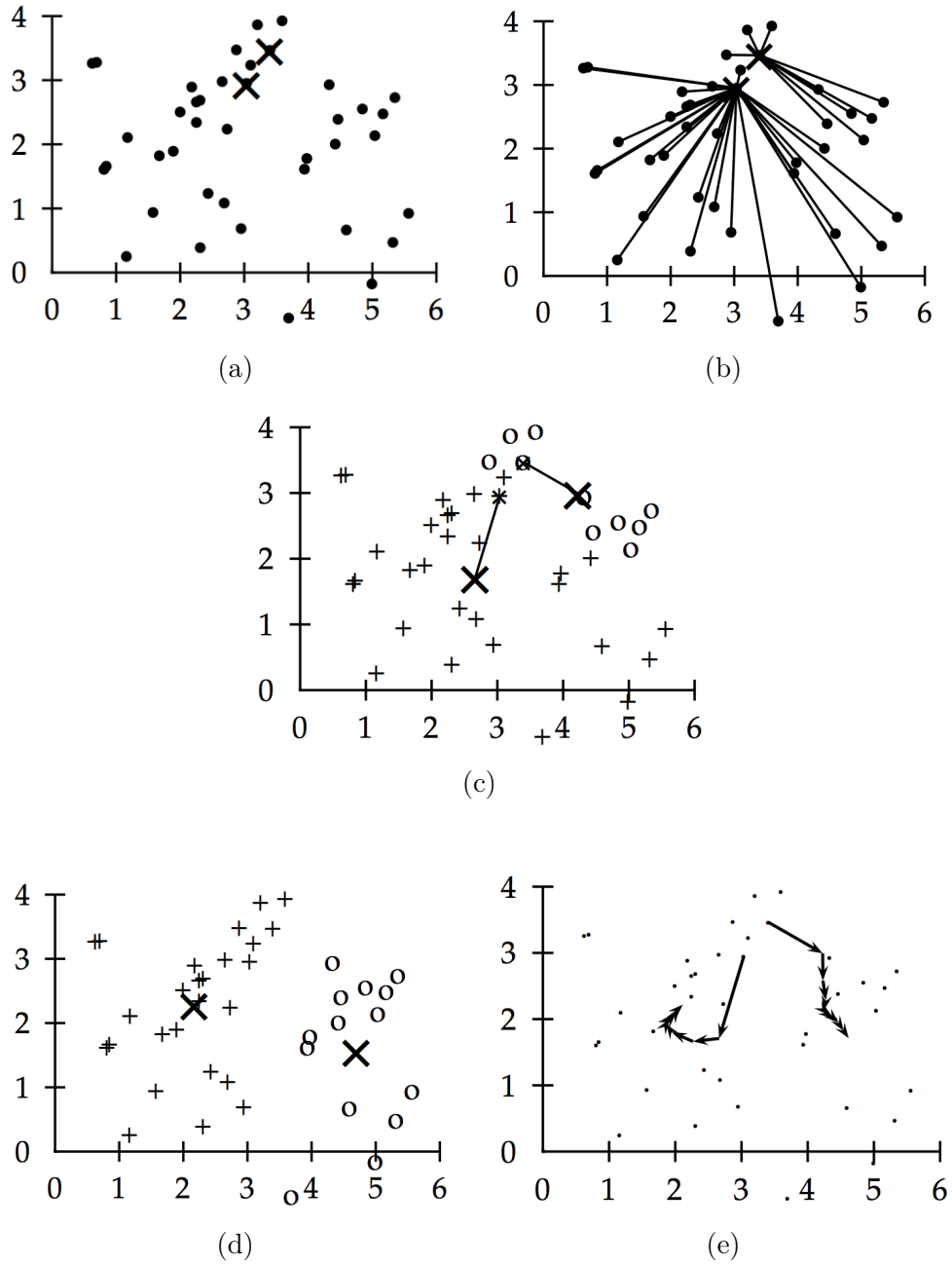

Figura 2.3: *K-means* aplicado a um problema do $\mathbb{R}^2$ com $K = 2$ (retirado de Manning et al. (2009)). (a) Inicialização dos centróides. (b) Atribuição dos elementos aos centróides. (c) Computação dos centróides. (d) Centróides após convergência. (e) Movimentos dos centróides em 9 iterações.

2.2 Discretização

Um método de discretização consiste basicamente em converter os valores de características – ou atributos – expressas de forma contínua em valores discretos. Segundo Catlett (1991) e Hwang e Li (2002), um método de discretização pode aumentar a precisão e velocidade de treinamento de um modelo classificador. Além disso, neste trabalho os métodos de discretização serão utilizados para a determinação das faixas de valores apresentadas em rótulos.

Conforme Kotsiantis e Kanellopoulos (2006), o objetivo de um método de discretização consiste em encontrar um conjunto de pontos de corte de modo a particionar uma faixa de valores contínuos em um conjunto de pequenos intervalos. Assim, um valor discreto será associado a cada intervalo diferente de valores contínuos.

Ainda conforme Kotsiantis e Kanellopoulos (2006), um ponto de corte se refere a um valor que divide um intervalo maior em outros dois menores de modo que parte do intervalo é menor ou igual ao ponto de corte e a outra parte é maior. Dessa forma, dado um intervalo $[a, b]$, um ponto de corte c divide a faixa de valores em duas partes: $[a, c]$ e $]c, b]$.

Na literatura existem diversos métodos de discretização e considerando apenas os métodos de discretização não-supervisionados apresentados em Kotsiantis e Kanellopoulos (2006), Cerquides e de Màntaras (1997) e Dougherty et al. (1995), os dois mais comumente utilizados são: Discretização por Larguras Iguais (*Equal Width Discretization - EWD*) e Discretização por Frequências Iguais (*Equal Frequency Discretization - EFD*).

2.2.1 Discretização por Larguras Iguais - *EWD*

Um outro método de discretização bastante utilizado é o *EFD* (Discretização por Frequências Iguais). Neste método a quantidade de elementos com valores distintos

entre os pontos de corte se mantém constante.

O método de Discretização por Larguras Iguais (EWD) pode ser aplicado calculando-se uma distância que deverá ser respeitada ao longo do intervalo. Dado a quantidade de valores R a serem gerados na discretização e um determinado intervalo $[a, b]$ representado por valores contínuos, o método EWD consiste em particionar o intervalo em R faixas de valores de tamanhos iguais. Assim, para gerar R intervalos serão necessários $R - 1$ pontos de corte. É importante ressaltar que os elementos pertencentes ao intervalo devem estar ordenados de forma crescente. A largura de cada faixa de valor $(r_1, ..., r_R)$ é representada por ω e pode ser calculada pela diferença entre os limites superior e inferior do intervalo dividido pela quantidade R de valores a serem gerados (2.10).

$$\omega = \frac{b - a}{R} \tag{2.10}$$

Uma vez que ω é calculado, podemos determinar os pontos de cortes $(c_1, ..., c_{R-1})$ que irão delimitar as faixas de valores. O primeiro ponto de corte, c_1, é dado pela soma do limite inferior a com a distância ω. Os próximos pontos de corte $(c_i, ..., c_R)$ podem ser calculados pela soma do ponto de corte anterior com ω (2.11).

$$c_i = \begin{cases} a + \omega, & \text{se } i = 1 \\ c_{i-1} + \omega, & \text{caso contrário} \end{cases} \tag{2.11}$$

Finalmente, os valores contínuos passarão a ser representados por i, onde i é o índice que indica a qual faixa r_i o valor se encontra. Por exemplo, para dividirmos o intervalo $[a, b]$ em R faixas de valores distintas precisaremos de $R-1$ pontos de corte (Figura 2.4). Qualquer valor pertencente ao intervalo $[a, c_1]$ terá um valor discreto associado igual ao índice de sua faixa r_1. Isto é, um valor que se encontra na faixa r_1 passará a ser representado pelo valor 1. De maneira análoga um valor que se encontra na faixa

$r_2 =]c_1, c_2]$ por 2 e, finalmente, um valor que se encontra em uma faixa qualquer r_i será representado por i.

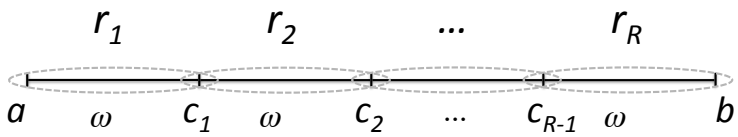

Figura 2.4: Discretização por *EWD*.

2.2.2 Discretização por Frequências Iguais - *EFD*

Um outro método de discretização bastante utilizado é o *EFD* (Discretização por Frequências Iguais). Neste método a quantidade de elementos com valores distintos entre os pontos de corte se mantém constante. Dado a quantidade de valores R a serem gerados por discretização, um determinado intervalo $[a, b]$ e a quantidade de elementos com valores distintos, ξ ($\xi \geqslant R$), ao longo deste intervalo a discretização por *EFD* irá particiona-lo em R faixas de valores que possuem a mesma quantidade de elementos distintos. $R - 1$ pontos de corte serão necessários para gerar R faixas de valores, $(r_1, ..., r_R)$, de modo que cada faixa contém a mesma quantidade λ de elementos distintos que pode ser calculada pelo valor inteiro da divisão entre a quantidade de elementos distintos e a quantidade de faixas de valores (2.12).

$$\lambda = \frac{\xi}{R} \tag{2.12}$$

Em alguns casos a má distribuição dos valores de um dado atributo podem ocasionalmente reunir uma grande quantidade de elementos com um mesmo valor e, portanto, causar um desequilíbrio na distribuição dos elementos em relação às faixas de valores.

Por esse motivo, este método de discretização considera apenas valores distintos em cada faixa, evitando que elementos com um mesmo valor sejam atribuídos a faixas diferentes.

Após o cálculo de λ e a ordenação dos elementos distintos existentes no intervalo $[a, b]$ em um vetor computacional contendo R elementos ($\nu_{[R]}$) podemos determinar os pontos de cortes $(c_1, ..., c_{R-1})$ que irão delimitar as faixas de valores. Cada ponto de corte pode ser definido como o valor do $(i \cdot \lambda)$-ésimo elemento ordenado no intervalo a partir de a. Isto é, cada ponto de corte c_i pode ser calculado por $v_{[i\lambda]}$ (2.13).

$$c_i = \nu_{[i\lambda]} \tag{2.13}$$

Finalmente, assim como no método anterior, os valores contínuos passarão a ser representados por i, onde i é o índice que indica a qual faixa r_i o valor se encontra. Portanto, para dividirmos o intervalo $[a, b]$ em R faixas de valores distintas precisaremos de $R - 1$ pontos de corte (figura 2.5). Qualquer valor pertencente ao intervalo $[a, c_1]$ terá um valor discreto associado igual ao índice de sua faixa r_1. Em outras palavras, um valor que se encontra na faixa r_1 passará a ser representado pelo valor 1 e de maneira análoga, um valor que se encontra na faixa $r_2 =]c_1, c_2]$ por 2. Finalmente, um valor que se encontra em uma faixa qualquer r_i será representado por i. Observe que, diferente do método anterior (*EWD*), as faixas de valores podem assumir tamanhos diferentes.

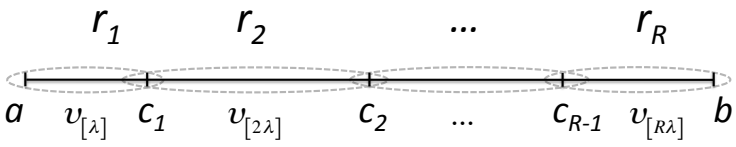

Figura 2.5: Discretização por *EFD*.

Capítulo 3

Rotulação: Modelo Proposto

Neste capítulo, inicialmente apresentam-se os problemas de agrupamento e rotulação. Alguns trabalhos relacionados são utilizados para a definição do problema e, em seguida, apresenta-se o modelo proposto presente neste trabalho composto de 4 etapas principais seguidas de suas respectivas explanações.

3.1 Problema

Conforme já apresentado no capítulo 2 e segundo Manning et al. (2009), o problema de agrupamento (*clustering*) consiste em definir um conjunto de grupos (*clusters*) que são coerentes internamente porém claramente diferentes entre si. Isto é, os elementos de um *cluster* devem ser tão similares quanto possível entre si e, ao mesmo tempo, tão diferentes quanto possível em relação ao elementos de outros *clusters*. Sendo um dos problemas mais conhecidos na área de aprendizagem não-supervisionada, o problema de *clustering* apresenta diversas soluções algorítmicas como *Cobweb* (Fisher, 1987), *Mapas auto-organizáveis de Kohonen* (Kohonen, 1982), *Fuzzy Clustering* (Sato e Sato, 1995), técnicas híbridas (Ramathilaga et al., 2011; Jiang e Li, 2009), além do *K-means*, já apresentado no Capítulo 2, entre outras.

Embora existam diversas pesquisas acerca do problema de *clustering* poucos trabalhos focam em rotular os *clusters* gerados. A rotulação de um *cluster* buscar resumir sua definição a fim de melhor compreendê-lo. Segundo Tzerpos (2001), em um esforço para maximizar o desempenho e precisão de algoritmos que lidam com esse problema, muitos pesquisadores desviaram-se do fato de que o principal objetivo era, a princípio, a compreensão dos grupos formados e não a satisfação de um critério abstrato, como por exemplo, a maximização dos graus de similaridade e dissimilaridade entre elementos intra e extra-*clusters* respectivamente. Tal compreensão se deve, principalmente, aos valores apresentados pelas características mais importantes de seus elementos. Assim, este conjunto de valores relevantes representa uma definição para um *cluster* qualquer – isto é, um *rótulo* – capaz de fornecer ao especialista um melhor entendimento sobre o mesmo.

De fato, são poucos os trabalhos dedicados à compreensão de *clusters*. O trabalho de Treeratpituk e Callan (2006) lida com *clusters* hierárquicos, de modo que cada *cluster* pode ser subdivido em *subclusters* de forma recursiva. Além disso, os rótulos são restritos a informações textuais. Conforme Treeratpituk e Callan (2006), embora existam muitos trabalhos relacionados a agrupamento hierárquico, poucos focam em defini-los. Nesses trabalhos, os descritores de *clusters* frequentemente falham em fornecer uma descrição compreensiva e em alguns casos fornecem apenas uma lista de termos que ainda necessitam ser avaliados por uma pessoa.

Ainda que árvores de decisão – *C4.5* e *ID3* (Quinlan, 1986), por exemplo – apresentem regras de classificação – específicas para o problema como um todo, mas não para os *clusters* de forma individual –, a extração de suas regras a fim de descrever um *cluster* específico pode ser bastante complexa, ou até mesmo inviável, uma vez que essas se encontram misturadas em várias condições envolvendo os diferentes valores de seus atributos. Por outro lado, classificar um elemento desconhecido conforme uma árvore de decisão é bastante simples já que basta verificar algumas condições, de forma

hierárquica, até encontrar um *cluster* a ser associado. Dessa forma, observe que (i) o problema de classificação pode nos ajudar a entender o problema como um todo mas não seus grupos de forma individual e que (ii) as árvores de decisão dificilmente apresentarão descritores para um grupo específico como a metodologia apresentada neste trabalho uma vez que cada uma lidam com problemas diferentes.

Embora existam outros trabalhos relacionados voltados para a rotulação de *clusters* hierárquicos (Glover et al., 2002; Chuang e Chien, 2004; Popescul e Ungar, 2000; Maqbool e Babri, 2005), todos eles se caracterizam por trabalhar exclusivamente com informações textuais. Alguns outros trabalhos (Chen et al., 2008; Eltoft e deFigueiredo, 1998) se referem à rotulação de *clusters* como o problema de atribuir um rótulo – isto é, um *cluster* – a um elemento desconhecido. Em outras palavras, se referem ao já conhecido problema de classificação. Portanto, nenhum trabalho foi encontrado envolvendo a rotulação de *clusters* no que se diz respeito em apresentar uma definição e obter conhecimento em relação a atributos numéricos relevantes. Isso posto, o problema de compreender *clusters* – ou ainda, o *problema de rotulação* – pode ser formalmente definido como segue.

Problema de Rotulação: *Dado um conjunto de clusters* $C = \{c_1, ..., c_K \mid K \geqslant 1\}$, *de modo que cada cluster contém um conjunto de elementos* $c_i = \{\vec{e}_1, ..., \vec{e}_{n^{(c_i)}} \mid n^{(c_i)} \geqslant 1\}$ *que podem ser representados por um vetor de atributos definidos em* $\mathbb{R}^m$ *e expresso por* $\vec{e}_j^{(c_i)} = (a_1, ..., a_m)$ *e ainda que* $c_i \cap c_{i'} = \{\emptyset\}$ *com* $1 \leqslant i, i' \leqslant K$ *e* $i \neq i'$*; o objetivo consiste em apresentar um conjunto de rótulos* $R = \{r_{c_1}, ..., r_{c_K}\}$ *no qual cada rótulo específico é dado por um conjunto de pares de valores, atributo e seu respectivo intervalo,* $r_{c_i} = \{(a_1, [p_1, q_1]), ..., (a_{m^{(c_i)}},]p_{m^{(c_i)}}, q_{m^{(c_i)}}])\}$ *capaz de melhor expressar o cluster* c_i *associado.*

A fim de esclarecimento, tem-se que:

- K é o número de *clusters*;
- c_i é um *cluster* qualquer;
- $n^{(c_i)}$ é o número de elementos do *cluster* c_i;
- $\vec{e}_j^{(c_i)}$ se refere ao j-ésimo elemento pertencente ao *cluster* c_i;
- m é a dimensão do problema;
- r_{c_i} é o rótulo referente ao *cluster* c_i;
- $]p_{m^{(c_i)}}, q_{m^{(c_i)}}]$[1] representa o intervalo de valores do atributo $a_{m^{(c_i)}}$ onde $p_{m^{(c_i)}}$ é o limite inferior e $q_{m^{(c_i)}}$ é o limite superior;
- $m^{(c_i)}$ é a quantidade de atributos presente em um rótulo referente ao *cluster* c_i.

Finalmente, o problema tem como entrada um conjunto de *clusters* e deve apresentar como saída um rótulo específico para cada grupo que melhor o define, conforme as especificações já apresentadas. Isso posto, a seguir apresenta-se um modelo criado como resolução desse problema.

3.2 Modelo

Dado o problema anteriormente definido, neste modelo apresenta-se uma abordagem capaz de rotular *clusters*, também apresentada em Lopes et al. (2013a) e Lopes et al. (2013b), desenvolvidos durante o decorrer deste trabalho.

Para contextualizar um problema real, no qual inicialmente tem-se apenas uma base de dados como entrada, um algoritmo com aprendizagem não-supervisionada foi introduzido em nossa metodologia e aplicado com o objetivo de formar vários grupos a partir dos elementos inicialmente fornecidos na base de dados. Para cada grupo formado

[1] Os limites dos intervalos devem ser ajustados conforme o modelo de discretização em aberto/fechado para que não haja coincidência de valores em intervalos distintos.

um segundo algoritmo, desta vez com aprendizagem supervisionada, será utilizado para a identificação de possíveis características importantes. Adicionalmente, faz-se uso de um método de discretização e de algumas estratégias de decisões, necessárias para a concretização desta abordagem subdividida em 4 principais etapas, apresentadas na Figura 3.1.

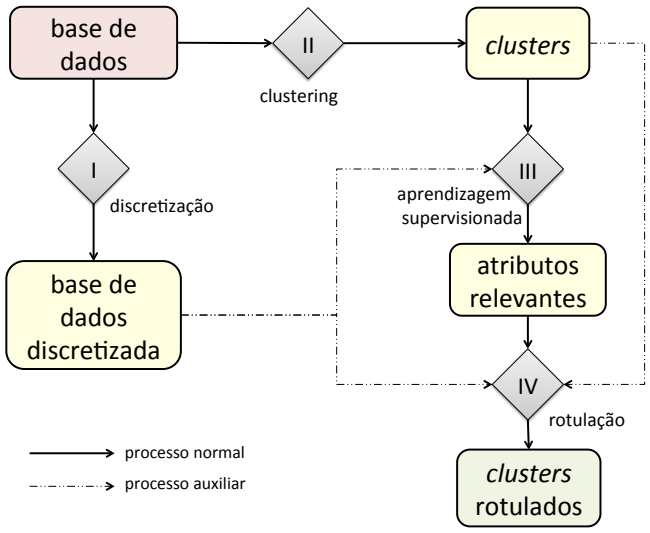

Figura 3.1: Modelo proposto.

Inicialmente, essa abordagem recebe como parâmetro de entrada uma base de dados. Essa base de dados pode conter diferentes tipos de dados[2] – discretos ou contínuos. Em *alguns casos* será necessário a aplicação de um método de discretização (I). O segundo passo (II) consiste em utilizar um algoritmo não-supervisionado para gerar os *clusters* a serem trabalhados. A base de dados discretizada não é utilizada nessa etapa mas sim a base de dados inicialmente fornecida. Em seguida (III), um algoritmo com aprendizagem supervisionada é aplicado para detectar quais são os atributos relevantes para a definição de cada grupo. Finalmente (IV), uma estratégia que seleciona o valor (para atributos discretos) ou intervalo de valor (para atributos contínuos) para cada atributo relevante selecionado é aplicada de forma a gerar rótulos.

[2]Outros tipos de dados podem ser facilmente representados por valores númericos.

A Tabela 3.1 representa uma base de dados, denominada Base de Dados Modelo (BDM), criada para facilitar o entendimento dessa abordagem. A primeira coluna contém um índice responsável por identificar cada elemento de forma única. As demais colunas representam uma característica – ou atributo – do respectivo elemento. Dessa forma, um elemento qualquer pode ser expresso como um vetor de dimensão m, onde m é a quantidade de características existentes na base de dados que o descrevem. Portanto, cada linha da tabela representa um elemento e cada coluna uma característica – exceto a primeira, que representa o índice. O elemento 3, por exemplo, pode ser descrito por $attr_1 = 2.00$, $attr_2 = 108.36$ e $attr_3 = 22.68$; ou seja, pelo vetor $\vec{e_3} = (2.00, 108.36, 22.68)$. Assim, dada uma base de dados como entrada a metodologia deste trabalho é detalhada em cada etapa a seguir.

3.2.1 Discretização – I

A etapa I consiste na discretização dos dados: atribuir valores discretos para os atributos que podem assumir uma grande variedade de valores dentro de um determinado domínio. Assim, o algoritmo com aprendizagem supervisionada utilizado na etapa III estará apto a identificar com menor complexidade um possível relacionamento entre os atributos, exibindo melhores resultados ao lidar com o problema de classificação envolvendo tais atributos. Conforme Catlett (1991) e Hwang e Li (2002), pode existir um aumento na acurácia e velocidade durante a etapa de treinamento ao se utilizar um método de discretização. Além disso, esse processo de discretização permite a inferência de uma faixa de valor, que acontece na etapa IV.

O processo de discretização se inicia com a escolha de quais atributos deverão ser discretizados bem como qual método – e seus devidos parâmetros – deverá ser devidamente aplicado a cada um, conforme as circunstâncias. Tais escolhas são de grande relevância para este trabalho e uma discussão acerca desse problema é apresentada no Capítulo 5.

Tabela 3.1: Base de Dados Modelo (BDM).

#	$attr_1$	$attr_2$	$attr_3$
1	2.08	92.11	22.07
2	1.26	85.03	20.45
3	2.00	108.36	22.68
4	1.74	43.78	18.72
5	1.82	100.20	23.09
6	1.43	77.59	21.80
7	1.53	44.01	20.98
8	1.14	107.77	18.99
9	1.97	98.00	22.32
10	1.50	39.67	21.78
11	1.74	55.86	20.31
12	1.80	65.72	19.62
13	1.33	82.01	19.82
14	1.66	103.93	21.10
15	1.42	66.14	21.61
16	1.87	88.36	22.45
17	1.11	107.82	19.32
18	2.08	67.66	20.74
19	1.85	82.65	20.35
20	1.04	102.62	19.46
21	1.97	100.37	21.94
22	1.95	45.70	22.10
23	1.77	50.04	20.16
24	1.97	81.57	19.83
25	1.52	93.13	20.61

#	$attr_1$	$attr_2$	$attr_3$
26	1.42	53.51	19.64
27	1.12	62.71	19.07
28	2.09	60.58	20.20
29	1.95	69.23	19.68
30	1.03	47.81	19.47
31	1.75	90.92	21.39
32	1.72	42.35	22.89
33	1.47	101.77	19.20
34	1.53	41.16	22.67
35	1.44	93.61	21.03
36	1.51	98.65	19.24
37	1.06	68.82	21.68
38	1.48	80.40	21.43
39	1.14	61.59	19.90
40	1.08	91.93	20.81
41	1.62	79.21	18.43
42	1.68	80.87	18.42
43	1.81	98.24	22.13
44	1.30	69.27	18.83
45	1.80	101.21	21.61
46	1.79	72.02	22.02
47	1.56	81.71	22.10
48	1.98	77.16	21.71
49	1.86	89.12	22.84
50	1.55	76.01	19.74

Assim, para este exemplo foram selecionados os atributos $attr_1$, $attr_2$ e $attr_3$ utilizando-se a técnica de discretização por frequências iguais (EFD) com $R = 3$ para a discretização dos mesmos. A Figura 3.2 mostra como ocorre a discretização nos atributos.

Assim, os valores iniciais são convertidos em valores discretos. Os elementos situados entre o valor mínimo e o primeiro ponto de corte passarão a ser representados pelo valor discreto 1; os valores situados entre os dois pontos de cortes pelo valor 2; e os valores entre o segundo ponto de corte e o valor máximo, pelo valor 3.

Como resultado obtém-se uma nova base de dados, neste exemplo denominada Base

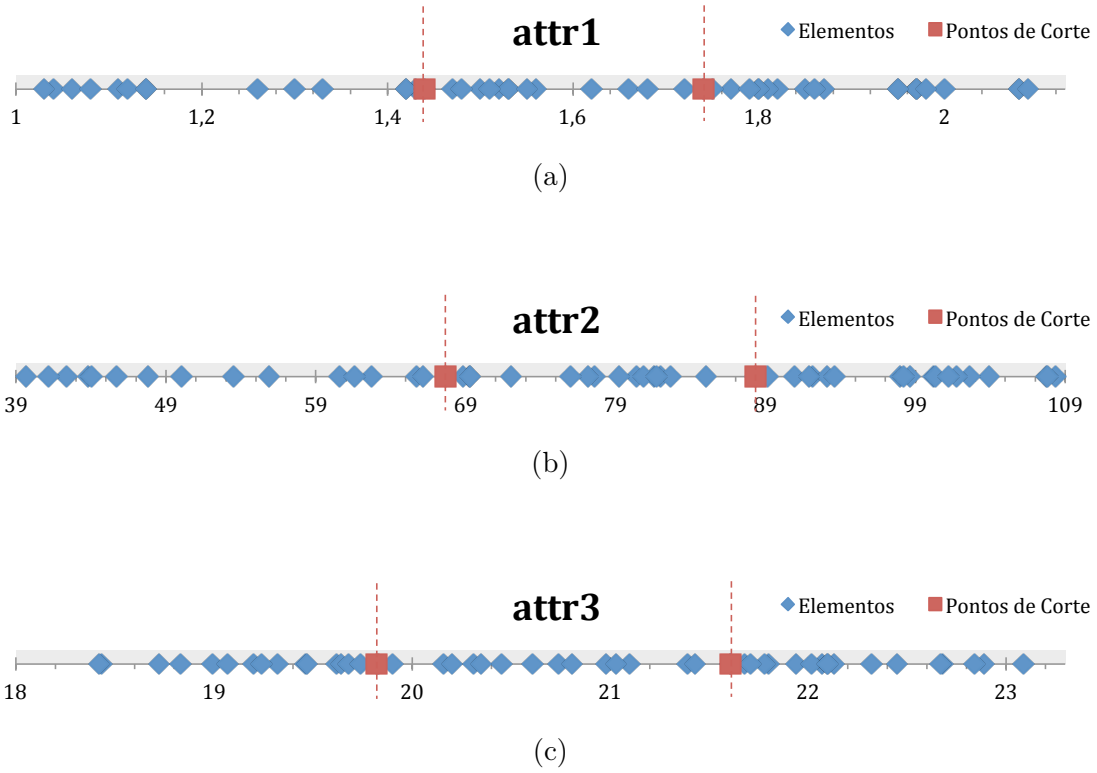

Figura 3.2: Discretização de atributos utilizando *EFD* com $R = 3$. (a) Discretização em $attr_1$. (b) Discretização em $attr_2$. (c) Discretização em $attr_3$.

de Dados Modelo Discretizada (BDMD), análoga à apresentada na Tabela 3.2. Esses novos valores obtidos com a discretização serão utilizados como entrada para o algoritmo com aprendizagem supervisionada na etapa III.

Assim, essa técnica permite o algoritmo lidar com faixas de valores representadas por valores discretos. Embora o processo resulte em uma representação do conhecimento de forma diferente – na qual existe perda de informação –, trabalhar com faixas de valores pode facilitar o aprendizado, especialmente na forma como será utilizado neste trabalho, com o objetivo de apresentar melhores resultados em termos de velocidade e precisão.

Tabela 3.2: Base de Dados Modelo Discretizada(BDMD).

#	$attr_1$	$attr_2$	$attr_3$
1	3	3	3
2	1	2	2
3	3	3	3
4	2	1	1
5	3	3	3
6	1	2	3
7	2	1	2
8	1	3	1
9	3	3	3
10	2	1	3
11	2	1	2
12	3	1	1
13	1	2	1
14	2	3	2
15	1	1	2
16	3	2	3
17	1	3	1
18	3	1	2
19	3	2	2
20	1	3	1
21	3	3	3
22	3	1	3
23	3	1	2
24	3	2	2
25	2	3	2

#	$attr_1$	$attr_2$	$attr_3$
26	1	1	1
27	1	1	1
28	3	1	2
29	3	2	1
30	1	1	1
31	3	3	2
32	2	1	3
33	2	3	1
34	2	1	3
35	1	3	2
36	2	3	1
37	1	2	3
38	2	2	2
39	1	1	2
40	1	3	2
41	2	2	1
42	2	2	1
43	3	3	3
44	1	2	1
45	3	3	2
46	3	2	3
47	2	2	3
48	3	2	3
49	3	3	3
50	2	2	1

3.2.2 Agrupamento (*Clustering*) – II

Após a discretização ocorre a geração dos *clusters* (etapa II). O problema de agrupamento apresenta diversas soluções na literatura – inclusive algumas já apresentadas no início deste capítulo – que recebem como entrada um conjunto de elementos (neste exemplo a base de dados BD) e apresenta como saída a associação de cada elemento a um respectivo *cluster* criado.

Para este exemplo – e testes utilizados neste trabalho – o *K-means* foi escolhido embora qualquer outro algoritmo com aprendizagem não-supervisionada que seja capaz

de lidar com o problema de *clustering* possa ser utilizado. Assim, a Tabela 3.3 mostra a BDM após o processo de agrupamento (*K-means*, com $K = 3$) contendo adicionalmente a coluna *cluster* que indica o grupo a qual um determinado elemento foi associado.

Tabela 3.3: Base de Dados Modelo (BDM) após agrupamento.

#	$attr_1$	$attr_2$	$attr_3$	*cluster*
1	2.08	92.11	22.07	2
2	1.26	85.03	20.45	1
3	2.00	108.36	22.68	2
4	1.74	43.78	18.72	3
5	1.82	100.20	23.09	2
6	1.43	77.59	21.80	1
7	1.53	44.01	20.98	3
8	1.14	107.77	18.99	2
9	1.97	98.00	22.32	2
10	1.50	39.67	21.78	3
11	1.74	55.86	20.31	3
12	1.80	65.72	19.62	1
13	1.33	82.01	19.82	1
14	1.66	103.93	21.10	2
15	1.42	66.14	21.61	1
16	1.87	88.36	22.45	2
17	1.11	107.82	19.32	2
18	2.08	67.66	20.74	1
19	1.85	82.65	20.35	1
20	1.04	102.62	19.46	2
21	1.97	100.37	21.94	2
22	1.95	45.70	22.10	3
23	1.77	50.04	20.16	3
24	1.97	81.57	19.83	1
25	1.52	93.13	20.61	2
26	1.42	53.51	19.64	3
27	1.12	62.71	19.07	1
28	2.09	60.58	20.20	1
29	1.95	69.23	19.68	1
30	1.03	47.81	19.47	3
31	1.75	90.92	21.39	2
32	1.72	42.35	22.89	3
33	1.47	101.77	19.20	2
34	1.53	41.16	22.67	3
35	1.44	93.61	21.03	2
36	1.51	98.65	19.24	2
37	1.06	68.82	21.68	1
38	1.48	80.40	21.43	1
39	1.14	61.59	19.90	1
40	1.08	91.93	20.81	2
41	1.62	79.21	18.43	1
42	1.68	80.87	18.42	1
43	1.81	98.24	22.13	2
44	1.30	69.27	18.83	1
45	1.80	101.21	21.61	2
46	1.79	72.02	22.02	1
47	1.56	81.71	22.10	1
48	1.98	77.16	21.71	1
49	1.86	89.12	22.84	2
50	1.55	76.01	19.74	1

3.2.3 Aprendizagem Supervisionada – III

Uma vez que os grupos estão devidamente formados e os dados a serem trabalhados, se necessário, já estão discretizados, inicia-se de fato o trabalho de rotulação. Com base na definição apresentada no início desse capítulo, cada rótulo referente a um *cluster*

qualquer é baseado é um conjunto de atributos e seus respectivos intervalos de valores. Assim, nessa etapa ocorre a detecção de quais atributos são relevantes para a definição de um *cluster* qualquer. Portanto, esta etapa possui como entrada um conjunto de *clusters* e apresenta como saída um conjunto de atributos para cada grupo gerado que serão utilizados em sua rotulação.

Para isso, serão utilizadas redes neurais artificiais do tipo PMC embora, a princípio, qualquer outro algoritmo com aprendizagem supervisionada capaz de detectar relação entre variáveis possa ser escolhido. Portanto, neste etapa, para cada *cluster* gerado serão criadas m RNAs, onde m é a quantidade de características que descreve um elemento do problema definido em $\mathbb{R}^m$. A Figura 3.3 apresenta RNAs aplicadas a um *cluster* qualquer fornecido por BDM após a tarefa de agrupamento.

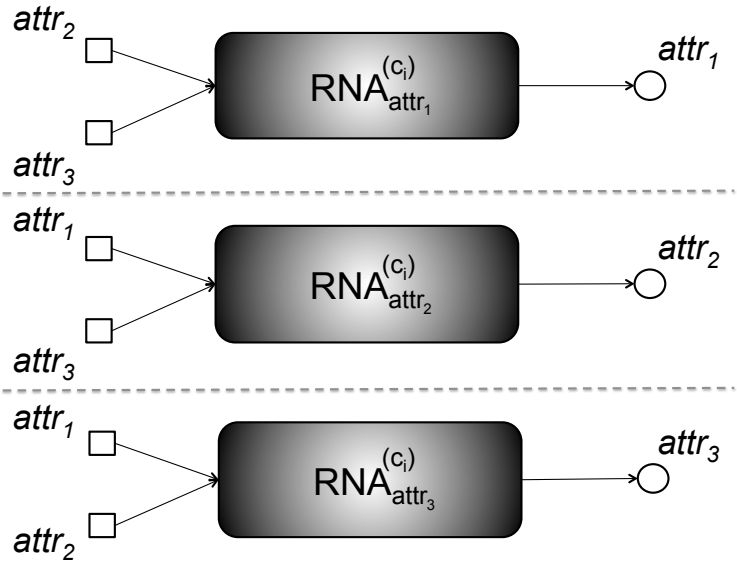

Figura 3.3: RNAs para seleção de atributos de um *cluster* qualquer em BDM após agrupamento.

Cada RNA criada será responsável por avaliar a relevância de um determinado atributo como um potencial candidato ao rótulo de seu grupo. A $RNA_{attr_1}^{(c_i)}$, por exemplo, é responsável por avaliar a importância do atributo $attr_1$ em relação ao *cluster* c_i. Assim,

dado um elemento n durante a etapa de treinamento, a saída desejada para a rede que avalia um atributo qualquer, *attr*, deverá ser o valor desse mesmo atributo apresentado em n. Para a rede $RNA_{attr_1}^{(c_i)}$, por exemplo, o valor desejado a ser apresentado é n_{attr_1}. Os demais atributos serão tratados como entrada da rede.

Dessa forma, busca-se, por intermédio de uma RNA, encontrar uma relação entre os atributos que são apresentados como entrada da rede e o atributo apresentado como saída. Caso exista uma relação, isso significa que o atributo avaliado pela RNA pode ser descrito ou representado pelos demais. Assim, para um determinado *cluster* o atributo avaliado é capaz de resumir as demais características do problema sendo, portanto, considerado um atributo relevante.

O quão relevante um atributo é – em relação a um determinado *cluster* – pode ser calculado conforme a porcentagem de acerto da RNA que o avalia durante a etapa de testes, após o treinamento. Assume-se então que a quantidade de acerto (dada em %) de uma rede indica se existe alguma relação entre os valores de entrada com o de saída: quanto maior o acerto pode-se dizer que maior foi a capacidade da rede em aprender sobre esta relação. É importante ressaltar que os valores utilizados como entrada e saída dessas RNAs são, quando existentes, os valores discretizados.

Dessa forma, cada *cluster* gerado possui um conjunto de m RNAs associadas. Cada RNA, por sua vez, apresenta uma taxa de acerto em relação ao seu aprendizado, realizado apenas com os elementos de seus respectivos *clusters*. Para dar uma maior precisão e confiança às taxas de acerto aplica-se o método de *holdout* (*random subsampling*), onde o processo é repetido de modo iterativo[3] por M vezes. Assim, para cada RNA calcula-se a média de suas taxas de acerto obtidas em M iterações. A Figura 3.4 mostra a média das taxas de acerto das RNAs utilizando a BDMD.

Em seguida, as redes referentes a um determinado *cluster* são ordenadas por ordem decrescente de suas taxas de erro criando um de *ranking*. Assim, os atributos

[3]Em uma determinada iteração, os elementos utilizados durante a etapa de treinamento em todas as RNAs de um mesmo *cluster* são os mesmos.

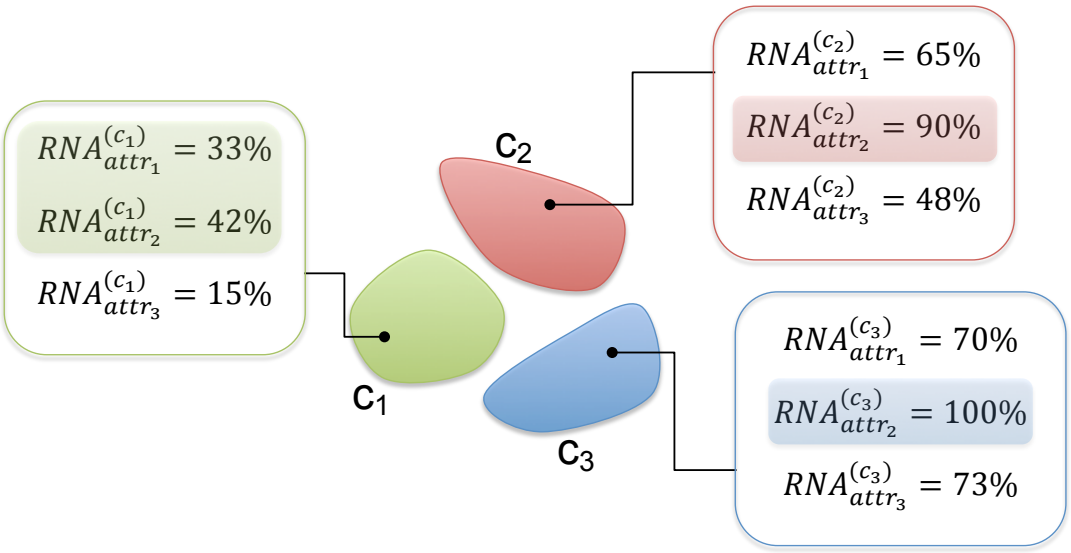

Figura 3.4: Médias das taxas de acerto de RNAs em BDMD.

relacionados às RNAs com maiores taxas de acerto são considerados os mais relevantes.

Embora *clusters* diferentes possam ter o mesmo conjunto de atributos relevantes espera-se que em pelo menos um desses atributos em comum exista uma diferença em relação aos seus intervalos de valores. Dessa forma, *clusters* diferentes podem ser distinguidos não só por características diferentes, mas também por apresentar seus valores de forma diferente. Caso contrário, não faria sentido atribuir elementos com as mesmas características a grupos distintos. Por isso, uma variação V, expressa em porcentagem (%), é utilizada para eliminar a possível ambiguidade entre *clusters* selecionando todos os atributos que possuem até um diferença V em relação ao atributo com maior taxa de acerto e descartar os demais.

Finalmente, cada *cluster* c_i contem um rótulo r_{c_i} contendo um conjunto de $m^{(c_i)}$ elementos. Para o exemplo adotado, temos que: $r_{c_1} = \{attr_1, attr_2\}$, $r_{c_2} = \{attr_2\}$ e $r_{c_3} = \{attr_2\}$. O valor de cada atributo é calculado na etapa seguinte.

3.2.4 Rotulação – IV

A última etapa dessa abordagem consiste em calcular os intervalos para os atributos selecionados na etapa anterior. Em busca de representar a maioria do grupo, os valores escolhidos para cada atributo são os de maior frequência no grupo. Para cada atributo a_j pertencente ao conjunto do rótulo r_{c_i} verifica-se qual valor possui a maior ocorrência dentro do *cluster* c_i. Para isso, utiliza-se a BDMD caso exista ou a BDM em caso contrário.

Por exemplo, considerando os atributos selecionados do rótulo r_{c_1}, os intervalos seriam definidos como $attr_1 = [1.03, 1.44]$ e $attr_2 =]67.66, 88.36]$. Isso porque o valor de maior ocorrência no *cluster* c_1 em relação ao atributo $attr_1$ é 1 e para o atributo $attr_2$ é 2. Assim, os limites dos intervalos dos atributos $attr_1$ e $attr_2$ são respectivamente $[p_1 = 1.03, q_1 = 1.44]$ e $]p_2 = 67.66, q_2 = 88.36]$, pois representam os limites das faixas de valores obtidas na discretização para os valores de maior frequência (Figura 3.2a) – 1 e 2 respectivamente. De maneira análoga, o mesmo ocorre com os atributos $attr_2$ e $attr_3$. Caso não houvesse discretização em um dos atributos selecionados o valor de maior ocorrência representaria o atributo. Observa-se que, nesse caso, não procura-se um de uma faixa de valores mas sim um valor específico. Assim, o intervalo formado por atributos que não passaram pelo processo de discretização possui seus limites inferior e superior iguais, com ambos representando o valor específico inferido.

Logo, os rótulos apresentados para os *clusters* são:

- $r_{c_1} = (attr_1, [1.03, 1.44]), (attr_2,]67.66, 88.36])$;
- $r_{c_2} = (attr_2,]88.36, 108.36])$;
- $r_{c_3} = (attr_2,]39.67, 67.66])$.

Alternativamente, a fim de simplificar, podemos escrever os rótulos da seguinte maneira: $r_{c_1} = attr_1[1.03 \sim 1.44]$ e $attr_2(67.66 \sim 88.36]$. Assim, entende-se que os

elementos do *cluster* c_1 possuem seu atributo $attr_1$ variando de 1.03 a 1.44 e seu atributo $attr_2$ variando de 67.66 a 88.36. Uma interpretação de maneira análoga serve para os demais atributos. Finalmente, o algoritmo 2 resume essa proposta em pseudocódigo.

Algoritmo 2: Rotulação

```
1:  Carregar base de dados;
2:  Selecionar e discretizar os atributos necessários;
3:  Executar o algoritmo não-supervisionado (clustering);
4:  for cada cluster do
5:      for cada iteração m = 1 até M do
6:          Definir conjuntos de treinamento e teste;
7:          for cada atributo do
8:              Executar treinamento do algoritmo supervisionado;
9:              Calcular a taxa de acerto;
10:         end for
11:     end for
12:     Calcular a média das taxas de acerto;
13:     Selecionar os atributos com maiores taxas de acerto em até V;
14:     Contar a maioria dos valores apresentados;
15:     Associar os valores aos intervalos;;
16: end for
17: Exibir rótulos;
```

Capítulo 4

Resultados

Neste capítulo, são apresentados os resultados obtidos conforme a implementação do modelo apresentado no Capítulo 3. Os resultados foram obtidos a partir de 4 base de dados distintas apenas com modificações em relação aos métodos de discretização utilizados e o parâmetro V, utilizados em suas execuções.

4.1 Detalhes de Implementação

O modelo proposto apresentado no Capítulo 3 foi implementado utilizando a ferramenta *MATLAB*[1]. Essa ferramenta possibilita a utilização de diversos algoritmos com aprendizagem supervisionada e não-supervisionada, dentre outras funções[2]. Além disso, a facilidade e robustez fornecida pelo ambiente pôde proporcionar testes iniciais que levaram ao melhor entendimento do problema abordado neste trabalho.

Em relação aos testes realizados neste trabalho, o *K-means* foi escolhido para lidar com a tarefa de agrupamento (*clustering*). O comando $kmeans(BD, K)$[3] foi utilizado onde BD corresponde à base de dados fornecida como entrada e K representa o número

[1] http://www.mathworks.com/products/matlab/
[2] Versão utilizada: R2012a (7.14.0.739), 64 bits (maci64).
[3] http://www.mathworks.com/help/stats/kmeans.html

de *clusters* a serem gerados. Os valores utilizados por tal comando não foram alterados obedecendo, então, aos padrões sugeridos pela ferramenta *MATLAB*. O valor de K não se demonstrou um problema pois as bases de dados utilizadas nestes testes fornecem, *a priori*, tal valor.

Ainda com relação aos testes apresentados neste capítulo, as RNAs foram escolhidas como método supervisionado para a seleção de atributos relevantes à rotulação. No *MATLAB* existem alguns diferentes tipos de implementações em relação às RNAs. A fim de simular uma rede PMC o comando $feedforwardnet()$[4] foi utilizado. Embora alguns testes tenham sido realizados com variações em relação à arquitetura da rede e ao método de treinamento os resultados apresentados foram obtidos com os valores padrões da ferramenta tanto para a RNA quanto para o método de treinamento utilizado. Na configuração padrão a rede possui 10 neurônios[5] em uma única camada oculta e o método de treinamento utilizado é o de retropropagação de Levenberg-Marquardt. A proporção de elementos para os conjuntos de treinamento e teste foram de 60% e 40% respectivamente.

Os parâmetros existentes na abordagem criada foram escolhidos após uma série de testes preliminares. O parâmetro $M = 10$ foi utilizado em todos os testes apresentados neste capítulo. Os métodos de discretização utilizados, bem como a quantidade de faixas de valores em cada atributo, e a variação V são especificados em cada base de dados. Os resultados obtidos em 4 bases de dados distintas são apresentados a seguir.

4.2 Identificação de Vidros

Esta base de dados se refere à identificação de vidros (*Identification Data Set Glass*) e pode ser encontrada no repositório de dados *UCI Machine Learning*[6] (Bache e Lichman,

[4] http://www.mathworks.com/help/nnet/ref/feedforwardnet.html

[5] Testes preliminares mostraram que configurações alternativas em relação ao número de camadas e/ou neurônios não demonstraram diferenças significativas.

[6] http://archive.ics.uci.edu/ml/

2013). O contexto de sua aplicação é na área forense onde a análise dos componentes de uma amostra de vidro pode identificar o tipo de vidro ajudando a solucionar crimes (Evett e Spiehler, 1988).

A base de dados contém 214 elementos (amostras de vidros), cada um representado por 9 atributos[7] com valores contínuos: o índice de refração (*IR*) e sua composição química, apresentados em porcentagem (*Na*, *Mg*, *Al*, *Si*, *K*, *Ca*, *Ba* e *Fe*). Os elementos podem ser agrupados em 7 diferentes tipos de *clusters*:

1. 70 elementos de janelas de construção (processado);
2. 76 elementos de janelas de construção (não processado);
3. 17 elementos de janelas de veículos (processado);
4. 0 elemento de janelas de veículos (não processado)[8];
5. 13 elementos de recipientes;
6. 9 elementos de utensílios de cozinha;
7. 29 elementos de faróis.

Neste caso, o método de discretização utilizado foi o *EWD* com $R = 4$ para todos os atributos; a variação é $V = 15$ e o agrupamento foi realizado com o *K-means* onde $K = 6$. Os resultados obtidos em relação a essa base de dados são apresentados na tabela 4.1.

A primeira coluna (*Cluster*) especifica um grupo; a segunda (*# Elem.*) indica a quantidade de elementos presentes no grupo; a terceira e quarta colunas, juntas, apresentam o rótulo indicado; a quinta coluna (*Rel. (%)*) indica a média da relevância do atributo – isto é, a média das taxas de acerto das redes de um determinado atributo

[7] O atributo classe (correspondente ao décimo atributo e responsável por identificar o tipo de vidro) foi removido da base de dados para a realização deste trabalho.

[8] Nenhum elemento desse tipo está presente na base de dados.

Tabela 4.1: Análise da rotulação para a base de dados de vidros.

		Rótulo			Análise	
Cluster	**# Elem.**	**Attr.**	**Faixa de Valores**	**Rel. (%)**	**# Erros**	**Acerto (%)**
1	74	Ba	$0 \sim 0.7875$	100	0	100
		K	$0 \sim 1.5525$	100	0	100
		Si	$72.61 \sim 74.01$	93.33	2	97.29
		Na	$12.3925 \sim 14.055$	90.33	3	95.94
2	5	Fe	$0 \sim 0.1275$	100	0	100
		Ca	$5.43 \sim 8.12$	100	0	100
3	19	K	$0 \sim 1.5525$	100	0	100
		Ba	$0 \sim 0.7875$	90	1	94.73
4	32	K	$0 \sim 1.5525$	100	0	100
		Ba	$0 \sim 0.7875$	93.07	1	96.87
		Ca	$8.12 \sim 10.81$	89.23	1	96.87
5	56	Ba	$0 \sim 0.7875$	100	0	100
		K	$0 \sim 1.5525$	100	0	100
		Na	$12.3925 \sim 14.055$	93.47	2	96.42
		Al	$1.0925 \sim 1.895$	88.69	4	92.85
		Mg	$3.3675 \sim 4.49$	85.65	6	89.28
6	28	Fe	$0 \sim 0.1275$	100	0	100
		K	$0 \sim 1.5525$	93.33	1	96.42

– em relação ao seu *cluster*. Finalmente, após a conclusão e apresentação dos rótulos para cada grupo é feita uma análise. Para cada elemento de um dado *cluster*, verifica-se se o valor real do atributo obedece ao rótulo sugerido. Assim, a sexta e sétima colunas apresentam, respectivamente, a quantidade de elementos que não se enquadram na definição apresentada e a porcentagem de elementos que obedecem ao rótulo sugerido, ambas em relação ao atributo da terceira coluna.

Observa-se que apenas os atributos dentro de uma variação V (15%) em relação ao atributo com maior relevância são apresentados. Assim, esse conjunto de atributos concatenados com seus valores representam o rótulo de um determinado grupo.

As colunas referente à rotulação mostram que 2 atributos (Ba e K) apresentaram uma relevância de 100% em relação ao *cluster* 1. Observa-se que considerando apenas esses 2 atributos o acerto em relação à rotulação é de 100%. Entretanto, os atributos

com maior relevância do *cluster* 5 também são Ba e K e com as mesmas faixas de valores apresentadas no *cluster* 1. Assim, existe um problema de ambiguidade – isto é, a definição para ambos grupos são a mesma – que pode ser resolvido considerando uma maior quantidade de atributos. Portanto, o preço a se pagar pela desambiguidade entre grupos é a confiança em atributos com menor relevância. Assim, considerando os atributos dentro de uma relevância aceitável, (V), os rótulos sugeridos contemplam todos os atributos conforme apresentado na tabela 4.1.

Outros *clusters* também apresentaram ambiguidade e em todos os casos, o parâmetro V foi capaz de solucionar esse problema. Observa-se as similaridades entre os atributos mais relevantes para os seguintes pares de grupos: 1-5, 2-6 e 3-4.

Mesmo que seja necessário a confiança em atributos com uma menor relevância os resultados obtidos apresentam altas taxas de acerto. Considerando os atributos menos relevantes, o *cluster* 5 apresentou a menor taxa de acerto em relação a esta base de dados de modo que apenas 6 elementos (de 56) não obedecem ao rótulo sugerido. Assim, em relação aos rótulos sugeridos, a menor taxa de acerto foi de 89.28% (atributo *Mg*, *cluster* 5) e a maior foi 100% (em vários atributos). Em média, os rótulos sugeridos foram capazes de classificar 95.54% dos elementos corretamente.

Finalmente, os rótulos apresentados são:

- $r_{c_1} = \{(Ba, 0 \sim 0.7875), (K, 0 \sim 1.5525), (Si, 72.61 \sim 74.01), (Na, 12.3925 \sim 14.055)\}$;
- $r_{c_2} = \{(Fe, 0 \sim 0.1275), (Ca, 5.43 \sim 8.12)\}$;
- $r_{c_3} = \{(K, 0 \sim 1.5525), (Ba, 0 \sim 0.7875)\}$;
- $r_{c_4} = \{(K, 0 \sim 1.5525), (Ba, 0 \sim 0.7875), (Ca, 8.12 \sim 10.81)\}$;
- $r_{c_5} = \{(Ba, 0 \sim 0.7875), (K, 0 \sim 1.5525), (Na, 12.3925 \sim 14.055), (Al, 1.0925 \sim 1.895), (Mg, 3.3675 \sim 4.49))\}$;

- $r_{c_6} = \{(Fe, 0 \sim 0.1275), (K, 0 \sim 1.5525)\}$;

4.3 Identificação de Sementes

A segunda base de dados utilizada nestes testes se refere à identificação de sementes de trigo (*Seeds Data Set*), também encontrada no repositório de dados *UCI Machine Learning*[9] (Bache e Lichman, 2013) e apresentada em Kulczycki e Charytanowicz (2011). Esse conjunto de dados se refere às sementes de 3 diferentes tipos de sementes de trigo onde cada amostra se refere a um tipo específico:

1. 70 elementos do tipo *Kama*;
2. 70 elementos do tipo *Rosa*;
3. 70 elementos do tipo *Canadian*.

Os 210 elementos podem ser descritos por 7 características[10] geométricas: área A, perímetro P, densidade C, comprimento da semente *(LK)*, largura da semente *(WK)*, coeficiente de assimetria AC, comprimento do sulco da semente LKG.

Neste caso, o método de discretização utilizado foi o *EFD* com $R = 3$ para todos os atributos; a variação é $V = 5$ e o agrupamento foi realizado com o *K-means* onde $K = 3$. Os resultados obtidos em relação a essa base de dados são apresentados na tabela 4.2.

Essa tabela segue o mesmo modelo da apresentada na subseção anterior. Portanto, são apresentados os *clusters*, a quantidade de elementos em cada grupo, os rótulos sugeridos, a relevância de cada atributo e uma análise, respectivamente, nas colunas da esquerda para a direita.

[9] http://archive.ics.uci.edu/ml/

[10] O atributo classe (correspondente ao oitavo atributo e responsável por identificar o tipo de trigo) foi removido da base de dados para a realização deste trabalho.

Tabela 4.2: Análise da rotulação para a base de dados de sementes trigos.

		Rótulo			**Análise**	
Cluster	**# Elem.**	**Attr.**	**Faixa de Valores**	**Rel. (%)**	**# Erros**	**Acerto (%)**
1	67	A	$12.78 \sim 16.14$	91.85	8	88.05
		P	$13.73 \sim 15.18$	87.03	9	86.56
2	82	A	$10.59 \sim 12.78$	92.42	12	85.36
		P	$12.41 \sim 13.73$	90.90	10	87.80
3	61	P	$15.18 \sim 17.25$	100	0	100
		WK	$3.465 \sim 4.033$	96.4	3	95.08
		LK	$5.826 \sim 6.675$	96.8	1	98.36
		A	$16.14 \sim 21.18$	100	0	100

Novamente, apenas os atributos dentro de uma variação V (5%) em relação ao atributo com maior relevância são apresentados. Assim, esse conjunto de atributos concatenados com seus valores representam o rótulo de um determinado grupo.

Diferente do caso anterior não houve o problema de ambiguidade nesses resultados, portanto um menor valor para V pôde ser utilizado. Observa-se que mesmo os atributos relevantes em comum aos grupos possuem faixas de valores diferentes. Os atributos A e P estão presentes em todos os 3 *clusters* e em cada um apresentam faixas de valores distintas.

O *cluster* 2 apresentou a menor taxa de acerto em relação a esta base de dados de modo que 12 elementos (de 82) não obedecem ao rótulo gerado. Assim, em relação aos rótulos sugeridos, a menor taxa de acerto foi de 85.36% (atributo A, *cluster* 2) e a maior foi 100% (em 2 atributos do *cluster* 3). Em média, os rótulos sugeridos foram capazes de classificar 89% dos elementos corretamente.

Finalmente, os rótulos apresentados são:

- $r_{c_1} = \{(A, 12.78 \sim 16.14), (P, 13.73 \sim 15.18)\}$;
- $r_{c_2} = \{(A, 10.59 \sim 12.78), (P, 12.41 \sim 13.73)\}$;
- $r_{c_3} = \{(P, 15.18 \sim 17.25), (WK, 3.465 \sim 4.033), (LK, 5.826 \sim 6.675), (A, 16.14 \sim 21.18)\}$;

4.4 Identificação de Plantas

A terceira base de dados utilizada nestes testes se refere à identificação de plantas *Iris* (*Iris Data Set*), também encontrada no repositório de dados *UCI Machine Learning*[11] (Bache e Lichman, 2013) e apresentada em Fisher (1936). Esse conjunto de dados contempla 3 diferentes tipos de plantas onde cada amostra se refere a um tipo específico de *Iris*:

1. 50 elementos do tipo *Iris-setosa*;
2. 50 elementos do tipo *Iris-versicolour*;
3. 50 elementos do tipo *Iris-virginica*.

Assim, os 150 elementos são representados por 4 características[12] com valores contínuos: comprimento da sépala (SL), largura da sépala (SW), comprimento da pétala PL e largura da pétala PW, todos expressos em centímetros.

Neste caso, o método de discretização utilizado foi o *EFD* com $R = 3$ para todos os atributos; a variação é $V = 10$ e o agrupamento foi realizado com o *K-means* onde $K = 3$. Os resultados obtidos em relação a essa base de dados são apresentados na tabela 4.3.

Tabela 4.3: Análise da rotulação para a base de dados de *Iris*.

		Rótulo			**Análise**	
Cluster	**# Elem.**	**Attr.**	**Faixa de Valores**	**Rel. (%)**	**# Erros**	**Acerto (%)**
1	50	PW	$0.1 \sim 1$	100	0	100
		PL	$1 \sim 3.7$	100	0	100
2	62	PL	$3.7 \sim 5.1$	89.2	6	90.32
3	38	PL	$5.1 \sim 6.9$	86.87	3	92.10
		PW	$1.7 \sim 2.5$	85.62	2	94.73

[11] http://archive.ics.uci.edu/ml/

[12] O atributo classe (correspondente ao quinto atributo e responsável por identificar o tipo de *Iris*) foi removido da base de dados para a realização deste trabalho.

Novamente, a tabela segue o mesmo modelo da apresentada na subseção anterior. Portanto, são apresentados os *clusters*, a quantidade de elementos em cada grupo, os rótulos sugeridos, a relevância de cada atributo e uma análise, respectivamente, nas colunas da esquerda para a direita.

Apenas os atributos dentro de uma variação V, dessa vez em (10%), em relação ao atributo com maior relevância são apresentados. Assim, esse conjunto de atributos concatenados com seus valores representam o rótulo de um determinado grupo.

Assim como no caso anterior a rotulação se demonstra bem definida. Observa-se que o atributo *PL* está presente em todos os *clusters* e que em cada um apresenta faixas de valores diferentes: $c_1(1 \sim 3.7)$, $c_2(3.7 \sim 5.1)$ e $c_3(5.1 \sim 6.9)$. Observa-se que mesmo sendo diferentes aos 3 grupos, a presença de outros atributos no rótulo é importante, pois também possuem uma relação com os demais atributos de modo a representá-los.

O *cluster* 2 apresentou a menor taxa de acerto em relação a esta base de dados de modo que 6 elementos (de 62) não obedecem ao rótulo apresentado. Assim, em relação aos rótulos sugeridos, a menor taxa de acerto foi de 90.32% (atributo *PL*, *cluster* 2) e a maior foi 100% (nos 2 atributos do *cluster* 1). Em média, os rótulos sugeridos foram capazes de classificar 94.14% dos elementos corretamente.

Finalmente, os rótulos apresentados são:

- $r_{c_1} = \{(PW, 0.1 \sim 1), (PL, 1 \sim 3.7)\}$;
- $r_{c_2} = \{(PL, 3.7 \sim 5.1)\}$;
- $r_{c_3} = \{(PL, 5.1 \sim 6.9), (PW, 1.7 \sim 2.5)\}$;

4.5 *Scientia.NET*

A quarta base de dados utilizada foi a *Scientia.Net*, apresentada em de Lima e Machado (2012). Esta base de dados se refere a um protótipo de rede social voltada para cientistas que desejam compartilhar suas pesquisas com outros pesquisadores. Além disso,

é dotada de algoritmos de aprendizagem de máquina que classificam seus usuários e conteúdo automaticamente.

O *Scientia.Net* possui 2000 usuários distribuídos em 20 áreas do conhecimento (100 amostras por área), caracterizados por 7 atributos[13] que indicam a principal área em diferentes níveis de escolaridade: graduação *Grad.*, mestrado *MSc.*, subárea mestrado *Sub MSc.*, doutorado *Dr.*, subárea doutorado *Sub Dr.*, pós-doutorado *Pós-doc.* e subárea pós-doutorado *Sub Pós-doc.*

As 20 áreas do conhecimento são representadas por números inteiros. A área de Geografia, por exemplo, é representada pelo número 27 nos atributos *Pós. Doc*, *Dr.* e *MSc.*; e pelo número 20 no atributo *Grad.*.

Neste caso, não se faz necessário uso de um método de discretização pois os atributos são discreto/categóricos. A variação utilizada foi $V = 5$ e o agrupamento foi realizado com o *K-means* onde $K = 20$. Os resultados[14] obtidos em relação a essa base de dados são apresentados na tabela 4.4.

Assim como nos exemplos anteriores se mantém o padrão da tabela exceto pela quarta coluna que apresenta um único valor e não uma faixa de valores. A única ambiguidade existente foi entre os *clusters* 3 e 20. Isso ocorreu devido ao agrupamento – 2 centróides próximos –, pois os *clusters* 3 e 20 possuem respectivamente 48 e 52 elementos onde ambos demonstraram 100% de acerto em seus rótulos; além de todos os atributos, inclusive os menos relevantes, serem idênticos assim como seus valores.

É importante observar que os rótulos são gerados em função dos *clusters* fornecidos. Portanto, grupos similares ou idênticos tendem a receber o mesmo rótulo quando na verdade deveriam representar um único *cluster*.

Existem outras consequências relacionadas diretamente ao agrupamento. *Clusters*

[13] O atributo classe (correspondente ao oitavo atributo e responsável por identificar a área de conhecimento do usuário) foi removido da base de dados para a realização deste trabalho.

[14] Apenas os *clusters* com as menores taxas de acerto e alguns com altas taxas foram selecionados para apresentação. Os demais grupos apresentaram taxas de acerto elevada e estão incluídos nas médias apresentadas posteriormente.

Tabela 4.4: Análise da rotulação para a base de dados *Scientia.Net*.

		Rótulo			**Análise**	
Cluster	**# Elem.**	**Attr.**	**Valor**	**Rel. (%)**	**# Erros**	**Acerto (%)**
...	...	...	...	...	...	...
2	12	Pós-doc.	17	100	0	100
		MSc.	17	100	0	100
		Dr.	17	100	0	100
		Sub Dr.	13	100	0	100
3	48	Pós-doc.	19	100	0	100
		Grad.	12	100	0	100
		MSc.	19	100	0	100
		Dr.	19	100	0	100
4	160	Pós-doc.	18	100	60	62.5
		MSc.	18	100	60	62.5
		Dr.	18	100	60	62.5
...	...	...	...	...	...	...
15	172	Pós-doc.	9	100	80	53.48
		Grad.	5	100	80	53.48
		MSc.	9	100	80	53.48
		Dr.	9	100	80	53.48
...	...	...	...	...	...	...
20	52	Pós-doc.	19	100	0	100
		Grad.	12	100	0	100
		MSc.	19	100	0	100
		Dr.	19	100	0	100

com poucos elementos como o *cluster* 2, por exemplo, tendem a possuir uma taxa de acerto mais elevada do que *clusters* que possuem uma quantidade de elementos acima do esperado, como ocorre nos *clusters* 4 e 15 – deveriam possuir apenas 100 elementos cada. Por serem representados por valores discretos, um grupo não pode representar ao mesmo tempo, duas áreas distintas. Ainda assim, mesmo que 80 elementos (*cluster* 15) não obedeçam ao rótulo sugerido, a maioria do grupo (53.48%) o faz.

Mesmo que muitos elementos não tenham obedecido ao rótulo, a média nesse caso ainda se manteve alta. Os menores valores foram 53.48% e 62.5%, encontrados nos *clusters* 15 e 4, respectivamente. Por outro lado, os *clusters* 1–3, 5, 8, 10, 13 e 17–20 apresentaram taxas de 100%. Os demais grupos apresentaram, em média, taxas

superiores a 92%. Finalmente, os rótulos sugeridos apresentaram uma definição coerente para 93.35% dos elementos.

Finalmente, alguns rótulos apresentados:

- $r_{c_3} = \{(Pos - doc., 19), (Grad., 12), (MSc., 19), (Dr., 19)\}$;
- $r_{c_4} = \{(Pos - doc., 18), (MSc., 18), (Dr., 18)\}$;
- $r_{c_15} = \{(Pos - doc., 9), (Grad., 5), (MSc., 9), (Dr., 9)\}$;

Capítulo 5

Conclusões

Neste capítulo são discutidos os pontos positivos e negativos da abordagem proposta no Capítulo 3. Adicionalmente, são listados alguns possíveis trabalhos futuros relacionados aos problemas encontrados.

Frente ao problema apresentado no capítulo 3, um algoritmo não-supervisionado foi utilizado para gerar *clusters* e, em seguida, aplicado um algoritmo com aprendizagem supervisionada em cada grupo formado para detectar quais atributos – e seus respectivos valores ou faixas de valores – podem ser utilizados para defini-los. Em outras palavras, o método apresentado busca apresentar um rótulo, capaz de definir um determinado grupo com base em suas características. Além disso, é importante destacar a utilização de um método de discretização nessa abordagem.

O principal propósito de utilizar um método de discretização consiste em permitir a inferência de um conjunto de valores para uma determinada característica de um rótulo. Dessa forma, um *cluster* não é limitado a ser representado por apenas um valor em um determinado atributo mas sim por um intervalo de valores.

O modo como foi utilizado nesse trabalho é algo a ser discutido. Em cada caso apresentado no capítulo 4 utiliza-se apenas um único método de discretização – com R

fixo – para todos os atributos da base de dados. Os atributos podem, e provavelmente possuem, diferentes distribuição de valores em seus respectivos domínios. A fim de obter melhores resultados, os métodos de discretização – e ajustes do parâmetro R – devem ser aplicados conforme a necessidade de cada atributo. Observe que quanto maior o valor de R menores serão os intervalos gerados, tornando-os mais especificos. Por outro lado, quanto menor o valor de R, maior o valor do intervalo. Poranto, deve-se encontrar um equilíbrio neste parâmetro a fim de melhor representar o grupo. Alternativamente, outras estratégias para a definição das faixas de valores – isto é, outras técnicas de discretização – podem ser empregadas.

A priori, um atributo deve ser selecionado para a discretização quando existe a necessidade de representá-lo por um conjunto de valores em vez de um único valor específico. Observe que essa escolha não restringe o atributo a ser do tipo contínuo embora seja necessário uma representação adequada do conhecimento.

É importante lembrar que o processo de rotulação é feito de acordo com os *clusters* fornecidos e que, portanto, nesse sentido, o algoritmo de agrupamento tem uma forte influência nos rótulos gerados. Assim, os rótulos podem variar em um mesmo problema conforme o conjunto de grupos apresentados em uma determinada instância do problema.

Embora exista uma grande quantidade de parâmetros que possam ser melhor ajustados (métodos de discretização para cada atributo, algoritmo não-supervisionado, algoritmo supervisionado, valores de R, V, M além das configurações específicas de cada método escolhido) a fim de obter uma melhoria significativa, os resultados se demonstraram bastante satisfatórios. A maioria dos *clusters* avaliados neste trabalho demonstraram altas taxas de acerto em relação aos rótulos sugeridos. A média de elementos que obedecem ao rótulo sugerido nos piores e melhores *clusters* de cada caso foi de 79,61% e 98.77%, respectivamente. A média de elementos correspondentes aos rótulos sugeridos envolvendo todos os *clusters* apresentados no Capítulo 4 foi de 93.50%.

Finalmente, algumas melhorias e alternativas ainda podem ser desenvolvidas:

- aplicar diferentes métodos de discretização – com diferentes valores para R – em cada atributo, conforme a necessidade exigida em cada um;
- criar um classificador a partir dos rótulos sugeridos;
- criar um classificador a partir das RNAs que detectaram os atributos relevantes em cada *cluster*;
- analisar os resultados obtidos com outras técnicas de aprendizagem de máquina como *Máquinas de Vetor de Suporte*, por exemplo.

Referências Bibliográficas

Bache, K. e Lichman, M. (2013). UCI machine learning repository.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. Em *Proceedings of the European Working Session on Learning on Machine Learning*, EWSL-91, pgs. 164–178, New York, NY, USA. Springer-Verlag New York, Inc.

Cerquides, J. e de Màntaras, R. L. (1997). Proposal and empirical comparison of a parallelizable distance-based discretization method. Em *In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*.

Chen, H.-L., Chuang, K.-T., e Chen, M.-S. (2008). On data labeling for clustering categorical data. *on Knowledge and Data Engineering, IEEE Transactions*, 20(11):1458–1472.

Chuang, S.-L. e Chien, L.-F. (2004). A practical web-based approach to generating topic hierarchy for text segments. Em *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pgs. 127–136. ACM.

de Lima, B. V. A. e Machado, V. P. (2012). Machine learning algorithms applied in automatic classification of social network users. *4th International Conference on Computational Aspects of Social Networks - CASoN*.

Dougherty, J., Kohavi, R., e Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. Em *ICML*, pgs. 194–202. Morgan Kaufmann.

Eltoft, T. e deFigueiredo, R. (1998). A self-organizing neural network for cluster detection and labeling. Em *IEEE International Joint Conference on Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence.*, volume 1, pgs. 408–412.

Evett, I. W. e Spiehler, E. J. (1988). Knowledge based systems. capítulo Rule induction in forensic science, pgs. 152–160. Halsted Press, New York, NY, USA.

Fisher, D. (1987). Improving inference through conceptual clustering. Em *Proceedings of the sixth National conference on Artificial intelligence - Volume 2*, AAAI'87, pgs. 461–465. AAAI Press.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188.

Glover, E., Pennock, D. M., Lawrence, S., e Krovetz, R. (2002). Inferring hierarchical descriptions. Em *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pgs. 507–514. ACM.

Han, J., Kamber, M., e Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

Haykin, S. (2001). *Redes neurais; princípios e prática*. Bookman, 2 edition.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.

Hwang, G. e Li, F. (2002). A dynamic method for discretization of continuous attributes. Em *Intelligent Data Engineering and Automated Learning — IDEAL 2002*, volume 2412 de *Lecture Notes in Computer Science*, pgs. 506–511. Springer Berlin Heidelberg.

Jiang, S.-Y. e Li, X. (2009). A hybrid clustering algorithm. Em *Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on*, volume 1, pgs. 366–370.

Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., e Wu, A. (2002). An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Em *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pgs. 1137–1143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kotsiantis, S. e Kanellopoulos, D. (2006). Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32:47–58.

Kulczycki, P. e Charytanowicz, M. (2011). A complete gradient clustering algorithm. Em *Proceedings of the Third International Conference on Artificial Intelligence and Computational Intelligence - Volume Part III*, AICI'11, pgs. 497–504, Berlin, Heidelberg. Springer-Verlag.

Levenberg, K. (1944). A method for the solution of certain problems in least squares. *Quart. Applied Math.*, 2:164–168.

Liu, H. e Yu, X. (2009). Application research of k-means clustering algorithm in image retrieval system. Em *Proceedings of the Second Symposium International Computer Science and Computational Technology(ISCSCT '09)*, pgs. 274–277.

Lopes, L. A., Machado, V. P., e Rabêlo, R. d. A. L. (2013a). Automatic labeling of groupings through supervised machine learning. Em *X Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, pgs. 1–6.

Lopes, L. A., Machado, V. P., e Rabêlo, R. d. A. L. (2013b). Clusters labeling through multi-layer perceptron algorithm. Em *XI Simpósio Brasileiro de Automação Inteligente (SBAI)*, pgs. 1–6.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. Em *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pgs. 281–297. University of California Press.

Manning, C. D., Raghavan, P., e Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Maqbool, O. e Babri, H. (2005). Interpreting clustering results through cluster labeling. Em *Proceedings of the IEEE Symposium on Emerging Technologies, 2005.*, pgs. 429–434.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441.

McCulloch, W. S. e Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Oyelade, O., Oladipupo, O., e Obagbuwa, I. (2010). Application of k-means clustering algorithm for prediction of students' academic performance. *(IJCSIS) International Journal of Computer Science and Information Security*, 7.

Popescul, A. e Ungar, L. H. (2000). Automatic labeling of document clusters.

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

Ramathilaga, S., Leu, J.-Y., e Huang, Y.-M. (2011). Adapted mean variable distance to fuzzy-cmeans for effective image clustering. Em *2011 First International Conference on Robot, Vision and Signal Processing (RVSP)*, pgs. 48–51.

Rodrigues, T. B., Macrini, J. L. R., e Monteiro, E. C. (2008). Seleção de variáveis e classificação de padrões por redes neurais como auxílio ao diagnóstico de cardiopatia isquêmica. *Pesquisa Operacional*, 28:285 – 302.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Sato, M. e Sato, Y. (1995). Fuzzy clustering model for fuzzy data. Em *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE Int*, volume 4, pgs. 2123–2128 vol.4.

Setiono, R. e Liu, H. (1997). Neural-network feature selector. *IEEE Transactions on Neural Networks*, 8(3):654–662.

Silva, I. N. d., Spatti, D. H., e Flauzino, R. A. (2010). *Redes neurais artificiais: para engenharia e ciências aplicadas*. Artliber.

Treeratpituk, P. e Callan, J. (2006). Automatically labeling hierarchical clusters. Em *Proceedings of the 2006 International Conference on Digital Government Research*, dg.o '06, pgs. 167–176. Digital Government Society of North America.

Tzerpos, V. (2001). *Comprehension-Drive Software Clustering*. PhD thesis, Universisity of Toronto.

Widrow, B. e Hoff, M. E. (1960). Adaptive switching circuits. Em *1960 IRE WESCON Convention Record, Part 4*, pgs. 96–104, New York. IRE.

Witten, I. H. e Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.