

Universidade Federal de Uberlândia - UFU

Faculdade de Computação - FACOM

Programa de Pós-graduação em Ciência da Computação

Geração de Regras de Decisão *Fuzzy* Utilizando a Teoria dos *Rough Sets*

Autor: Jean Carlo de Sousa Santos

Orientadora: Profa. Dra. Denise Guliato

Uberlândia – novembro de 2009



**FACULDADE DE COMPUTAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



Jean Carlo de Sousa Santos

Geração de Regras de Decisão *Fuzzy* Utilizando a Teoria dos *Rough Sets*

Dissertação de Mestrado apresentada à Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.
Área de concentração: Banco de Dados.

Orientadora:
Profa. Dra. Denise Guliato

Uberlândia – novembro de 2009

Agradecimentos

Agradeço primeiramente a Deus pela saúde e pela rara oportunidade de poder concluir mais este passo de minha caminhada.

À minha orientadora Profa. Denise Guliato, um exemplo de pessoa, meus mais sinceros e intensos agradecimentos pela enorme contribuição para o meu crescimento científico e pelos momentos de paciência, confiança e dedicação.

Aos meus pais, Marcos Antônio dos Santos e Vilma Andrade de Souza Santos pela oportunidade, pela educação e pela vida.

Aos meus irmãos Marcelo e Gabriel pela eterna parceria e companherismo.

Aos meus avós Luzia e Edgar e à minha tia Vádia pela espontânea dedicação. Ao meu tio Vilmar pela prestatividade.

À minha namorada Fernanda pelo amor e pelos momentos de incentivo e dedicação. A Vera pela sinceridade e carinho e a Flávia pelas longas conversas.

À ilustre amiga Elaine Faria que contribuiu fortemente para o meu crescimento. Ao Prof. Foued Espíndola, a TODOS os amigos do antigo LCC (Laboratório de Computação Científica), aos amigos do LBD (Laboratório de Banco de Dados), ao Dr. Donizete, ao Dr. Túlio Macedo e aos colegas do programa de pós-graduação deixo aqui a minha gratidão.

Aos amigos Rubens Samuel, Danilo Medeiros e Lígia.

À todos aqui mencionados ou não que diretamente ou indiretamente contribuíram para a realização deste trabalho, deixo meus recíprocos sentimentos e meu muito obrigado.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Resumo

Este trabalho propõe um novo método para gerar automaticamente regras *fuzzy* baseado na teoria dos *rough sets* e na teoria dos conjuntos *fuzzy*. As regras derivadas são concisas em relação ao número de termos antecedentes e apresentam alta taxa de cobertura. O sistema de classificação baseado nestas regras *fuzzy* foi adaptado para discriminar entre a possibilidade e a impossibilidade de classificação. A proposta foi testada em cinco bases de dados públicas fornecidas pela Universidade de Wisconsin, são elas: a Iris, a Wine, a *Wisconsin Diagnosis Breast Cancer* considerando 10 atributos (Wdbc com 10 atributos), a *Wisconsin Diagnosis Breast Cancer* considerando 30 atributos (Wdbc com 30 atributos) e a Wisconsin Prognostic Breast Cancer (Wpbc). A precisão de classificação obtida para Iris, Wine, Wdbc com 10 atributos, Wdbc com 30 atributos e Wpbc foram 100%, 100%, 99,03%, 98,03 e 93,90%, respectivamente.

Palavras-chave: Regras *Fuzzy*, *Rough Sets*, Reconhecimento de Padrões, Aprendizado de máquina, Classificação de padrões.

Abstract

This paper proposes a new method to automatically generate fuzzy rules based on rough sets theory and fuzzy sets theory. The derived rules are concise with respect to the number of antecedent terms and present high coverage rate. The classifier system based on these fuzzy rules was tailored to discriminate between possibility of classification and impossibility classification. The proposal was tested with five public databases provided by University of Wisconsin: the Iris, the Wine, the *Wisconsin Diagnosis Breast Cancer* considering 10 attributes (Wdbc with 10 attributes), the *Wisconsin Diagnosis Breast Cancer* considering 30 attributes (Wdbc with 30 attributes) and the Wisconsin Prognostic Breast Cancer (Wpbc). The classification accuracies obtained for Iris, Wine, Wdbc with 10 attributes, Wdbc with 30 attributes and the Wpbc were 100%, 100%, 99,03%, 98,03 e 93,90%, respectively.

Keywords: Fuzzy Rules, Rough Sets, Pattern Recognition, Machine Learning, Pattern Classification Task.

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Caracterização do problema	1
1.3 Objetivos	4
1.4 Organização do trabalho	5
2 Classificação de padrões baseada em regras de decisão	6
2.1 Introdução	6
2.2 Sistemas de classificação baseados em regras	7
2.3 Sistemas de classificação utilizando Teoria dos Rough Sets	11
2.3.1 Conceitos básicos sobre Teoria dos <i>Rough Sets</i>	12
2.3.2 Função de pertinência <i>rough</i>	14
2.3.3 Trabalhos relacionados	17
2.4 Discussão sobre os trabalhos relacionados	20
2.5 Considerações finais do capítulo	22
3 Proposta de um método para geração automática de regras	23
3.1 Introdução	23
3.2 Proposta de uma nova aproximação	24
3.3 Geração de regras fuzzy utilizando <i>upper</i> ^{α}	24
3.3.1 Pré-processamento	25
3.3.2 Obtenção de regras de decisão <i>fuzzy</i> - DM	26
4 Avaliação Experimental	31
4.1 Materiais e métodos	31
4.1.1 Bases de dados	31
4.1.2 Técnica de validação	32
4.2 Resultados	32

4.2.1	Comparação com outros trabalhos	32
4.2.2	Impacto de redução de dimensionalidade no método proposto e em outros métodos	34
4.2.3	Discussão dos resultados	36
5	Conclusão	38
5.1	Principais contribuições	38
5.2	Publicações	38
5.3	Trabalhos futuros	38
	Referências bibliográficas	40

Lista de Figuras

1.1	Exemplos de conjuntos gerados pelas aproximações tradicionais <i>lower</i> e <i>upper</i> e pela aproximação <i>upper^{alpha}</i> proposta.	4
2.1	Exemplo de <i>tabela de decisão</i>	12
2.2	Conjuntos elementares para a <i>tabela de decisão</i> da Figura 2.1 , considerando $B = \{\text{cor, tamanho}\}$	13
2.3	Conjuntos de aproximações <i>lower</i> e <i>upper</i> e regiões de interesse para a classe $X_1 : B_*(X_1) = \{x_1, x_2, x_3, x_5\}; B^*(X_1) = \{x_1, x_2, x_3, x_4, x_5, x_9\}$	15
2.4	Conjuntos de aproximações <i>lower</i> e <i>upper</i> e regiões de interesse para a classe $X_2 : B_*(X_2) = \{x_6, x_7, x_8, x_{10}\}; B^*(X_2) = \{x_4, x_9, x_6, x_7, x_8, x_{10}\}$	16
2.5	Algoritmo proposto por Sarkar.	18
3.1	Partição <i>fuzzy</i> com $p = 4$	25

Lista de Tabelas

3.1	Exemplo de tabela de decisão	27
3.2	<i>Rough sets</i> para as classes X_1 e X_2 considerando $B_1, B_2, B_3, B_4, B_5, B_6$ e B_7	29
4.1	Características das bases de dados utilizadas	32
4.2	Número de funções de pertinência <i>fuzzy</i> (trapézio), p , utilizadas para categorização dos <i>atributos condicionais</i> das bases de dados	32
4.3	Comparação do classificador proposto com outros métodos.	33
4.4	Resultados em termos de precisão de classificação, número médio de não classificáveis, número médio de termos antecedentes nas regras.	34
4.5	Resultados em termos de precisão de classificação relativa a base de dados Iris para o Teste 2 e Teste 3.	35
4.6	Resultados em termos de precisão de classificação relativa a base de dados Wine para o Teste 2 e o Teste 3.	35
4.7	Resultados em termos de precisão de classificação relativa a base de dados Wdbc com 10 características para o Teste 2 e o Teste 3.	35
4.8	Resultados em termos de precisão de classificação relativa a base de dados Wdbc para o Teste 2 e o Teste 3.	36

Introdução

1.1 Considerações Iniciais

O Homem classifica desde o princípio da humanidade. Na Grécia antiga com Aristóteles (384- 322 a.C.) [36] surgiu a primeira tentativa conhecida de classificação, na qual os animais eram divididos em dois grandes grupos: os com sangue e os sem sangue. Desde então, o ser humano sempre classificou as "coisas" porque isso as torna mais fáceis de serem compreendidas. Associar características de objetos a uma classe ou inferir uma classe a partir de características de objetos é uma forma de tornar o universo mais compreensível.

Com os avanços tecnológicos voltados para coleta, armazenamento e disponibilização de dados, há um crescente aumento na quantidade de informação implícita armazenada em bases de dados. Informação vem da palavra latina *informare*, que significa "dar forma". Deste modo, informação é o conjunto de dados que os seres humanos deram forma para torná-los significativos e úteis. Obter informações a partir de bases de dados volumosas é uma tarefa cara e demorada para ser realizada manualmente, sendo muitas vezes infactível devido à capacidade cognitiva humana. Neste contexto, dados coletados e armazenados podem tornar-se verdadeiros "túmulos de informação", pois jamais são revistos e/ou analisados. Emerge então a necessidade do desenvolvimento de técnicas e ferramentas computacionais capazes de auxiliar na extração de informações úteis contidas nestes grandes volumes de dados. Estas informações úteis quando presentes em um contexto que envolve a tarefa de classificação [56] podem ser adequadas para a construção de um classificador de padrões. Um classificador, após construído, é capaz de concluir algo a partir das informações extraídas e representa portanto um conhecimento. Este classificador, obtido a partir destas informações extraídas da base de dados, pode ser visto como um modelo final de um processo de descoberta de conhecimento em bases de dados - KDD (*Knowledge Discovery on Databases*) [38].

1.2 Caracterização do problema

A KDD atrai pesquisadores de diversas áreas, pois faz uso de conceitos de banco de dados, métodos estatísticos, ferramentas de visualização, técnicas de inteligência artificial (IA) e mi-

neração de dados (DM) ou *data mining* [56], [47]. O processo de KDD envolve 3 etapas, são elas :

1. Pré-processamento - Esta etapa verifica inconsistências nos dados, realiza o tratamento amostras com valores perdidos e transforma os dados de modo a modificar o espaço característica (seleção e/ou transformação do espaço característica) com o objetivo de preparar os dados para posterior aplicação de técnicas de DM.
2. DM - são aplicadas técnicas que fazem frente com as teorias de aprendizagem de máquina, estatística, agrupamento e modelos gráficos, para a classificação de padrões nos dados (extração de conhecimento).
3. Avaliação e interpretação do conhecimento extraído - etapa importante para verificação da precisão e da interpretabilidade do conhecimento extraído na etapa de DM.

Dentro da etapa de DM, o reconhecimento de padrões em grandes bases de dados está relacionado com um tópico que é a Aprendizagem de Máquina (AM) ou *Machine Learning* [56]. A AM faz uso de métodos para que o computador "aprenda" por meio de indução ou dedução. Métodos baseados em dedução (também chamada por Aristóteles de silogismo) inferem uma conclusão à partir de verdades já conhecidas, ou seja, parte do geral para o particular, por exemplo: Todos boxeadores usam luvas, João é um boxeador, logo João usa luvas. Em contrapartida, métodos indutivos criam explicações à partir da observação de fatos, ou seja, argumentam do particular para o geral, por exemplo: João usa luvas, João é um boxeador, logo alguns boxeadores usam luvas.

O método dedutivo não é muito utilizado para extração de conhecimento, pois quase sempre faltam premissas universais verdadeiras (por exemplo: boxeadores usarem luvas e João ser boxeador), e além disso um método dedutivo não extrai conhecimento, ele apenas infere a partir de conhecimentos já existentes.

O método indutivo está associado à metodologia científica observatória. Há estudiosos que, baseados na impossibilidade de se ter uma observação completa e universal, afirmam que não se pode ter certeza que chegou-se a uma verdade via o método indutivo. Contudo, podemos afirmar que para o material (domínio do problema abordado) utilizado, as conclusões são verdadeiras. Como este trabalho está em um contexto no qual não temos premissas verdadeiras sobre o domínio do problema, vamos utilizar um raciocínio indutivo para a construção/extração do conhecimento (etapa 4 - DM).

A AM utilizando o raciocínio indutivo ocorre de forma supervisionada e não supervisionada. Na supervisionada, o algoritmo de aprendizado é abastecido com uma base de dados com informações sobre as características das amostras e as classes as quais as amostras pertencem. Em contrapartida, na AM não supervisionada a informação prévia sobre as classes das amostras não existe. Os estudos desta dissertação estão contidos em uma AM supervisionada.

A AM para a tarefa de classificação envolve tópicos como árvores de decisão, redes neurais, algoritmos genéticos (AG), classificação bayesiana, regras de associação, regras de decisão, clusterização e outros [56]. Todos estes métodos armazenam relações descobertas entre as características das amostras sendo avaliados e suas possíveis classes. A principal vantagem da classificação baseada em regras com relação a outros modelos não lineares é a sua alta

interpretabilidade [21], pois a inferência realizada para classificação não segue um modelo "caixa preta" e pode ser entendida, sendo esta a motivação para seu uso.

Na maioria dos problemas de classificação e controle [43] do mundo real, a incerteza quanto a caracterização e consequente categorização das amostras está presente. Afim de representar as incertezas Zadeh [35] introduziu a teoria dos conjuntos *fuzzy*. Esta teoria é empregada neste trabalho para tal fim, portanto, as regras if-then geradas são *fuzzy*. É importante ressaltar que a propriedade de incerteza é relativa às amostras de um conjunto, e não ao conjunto como um todo, ou seja, há uma incerteza com relação ao grau de pertinência de uma amostra a um conjunto de amostras. Isto pode ser claramente observado por exemplo na dificuldade que temos ao tentar diferenciar dois irmãos gêmeos idênticos (univitelinos [9]). Nesta situação, devido a pouca dissimilaridade que existe quanto as suas características físicas individuais, podemos ficar incertos quando tentamos diferenciar visualmente um irmão do outro.

Uma propriedade do conjunto como um todo é a "vagueza" (*vagness*). Em problemas do mundo real esta propriedade se faz presente quando duas ou mais amostras que são similares com relação a um espaço característica, porém, pertencentes a classes diferentes, compõem um mesmo conjunto. Conjuntos com a propriedade de vagueza foram definidos por Pawlak e denominados como *Rough Sets* (RS) [66]. Pawlak definiu duas aproximações de conjuntos que podem ser utilizadas para informar quais amostras podem ser classificadas com certeza e quais podem possivelmente ser classificadas em uma dada classe C . O conjunto que com certeza contém apenas amostras que pertencem a uma determinada classe é aproximado pelo conjunto gerado pela aproximação denominada *lower*. O conjunto de amostras que possivelmente podem ser tidas como pertencentes a uma classe determinada é aproximado pelo conjunto gerado pela aproximação denominada *upper*.

A aproximação *lower* é muito restritiva e pode gerar um conjunto com poucas amostras classificáveis, pois, utiliza apenas aqueles RS compostos por amostras de uma mesma classe C sendo aproximada (conjuntos sem vagueza alguma). Já a aproximação *lower* é demasiadamente permissiva e pode gerar um conjunto pouco interessante para identificar amostras classificáveis, pois é necessário que apenas uma amostra no RS seja da classe sendo aproximada (conjuntos com alto grau de vagueza). Para superar estas limitações, este trabalho propõe uma extensão para as aproximações definidas na Teoria dos *Rough Sets*, denominada *upper^{alpha}*. A extensão proposta identifica com mais eficiência (pois não é tão restritiva nem tão permissiva) e com mais eficácia (pois é agora utilizado apenas um conjunto de amostras obtido de apenas uma aproximação e isto requer menor espaço de armazenamento) os RS que contém amostras classificáveis. Esta extensão gera um conjunto de amostras classificáveis mais adaptado para ser utilizado no processo de geração de regras *fuzzy* if-then proposto. As regras geradas por este processo compõem o núcleo de um sistema de classificação. A Figura 1.1 exemplifica, para um problema que envolve amostras pertencentes a duas classes, os conjuntos gerados pelas aproximações tradicionais definidas na TRS e o conjunto de amostras gerado fazendo uso da aproximação *upper^{alpha}*.

Uma classificação tradicional com dois estados conclui que uma amostra desconhecida y pertence ou não pertence a uma classe D . Em problemas de classificação do mundo real, as fronteiras entre as classes pertencentes ao domínio do problema podem sobrepor-se. Surge então uma incerteza quanto a afirmação de que y pertence ou não a uma das classes. Esta incerteza é representada por graus de pertinência *fuzzy* definidos no intervalo $[0, 1]$. Devido a falhas no

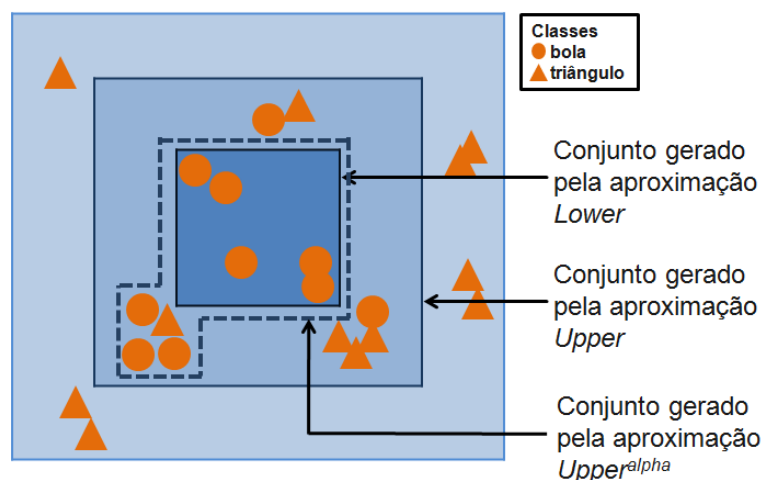


Figura 1.1: Exemplos de conjuntos gerados pelas aproximações tradicionais *lower* e *upper* e pela aproximação *upper^{alpha}* proposta.

processo de extração de características, uma amostra pode ser ambigualmente representada, indicando que pertence igualmente a duas classes diferentes ou não apresentar evidências quanto a pertinência a alguma das classes possíveis. Neste caso ocorre uma impossibilidade de classificação por igualdade de evidências ou ignorância de fatos.

Motivado pela existência de falta de evidências para classificação, em problemas do mundo real, este trabalho distingue a evidência e a ignorância no processo de classificação. Para tanto, caso uma amostra desconhecida y não tenha evidências para ser classificada em uma das classes possíveis, definidas no domínio problema, ela é classificada em uma classe especial "necessita de complemento". Esta distinção entre a evidência e a ignorância na classificação de y é importante em muitas situações, como por exemplo em diagnósticos médicos, em que ocorre a falta de informações para concluir um diagnóstico decisivo, ou seja, os sintomas são ambíguos ou não são característicos de alguma enfermidade.

1.3 Objetivos

O objetivo deste trabalho é utilizar o processo KDD para geração automática de regras *fuzzy* do tipo if-then, com o propósito de classificação de padrões. O método proposto garante que as regras sejam :

1. Regras concisas (número pequeno de termos antecedentes);
2. Número reduzido de regras;
3. Alta taxa de precisão de classificação.

Outro objetivo é propor um sistema de classificação de padrões baseado nestas regras geradas que seja capaz de distinguir entre a evidência e a ignorância na classificação de padrões.

1.4 Organização do trabalho

O Capítulo 2 apresenta uma revisão da literatura sobre os trabalhos relacionados relevantes, o Capítulo 3 apresenta todas as etapas da proposta para a geração de regras *fuzzy* if-then que compõem o sistema de classificação, o Capítulo 4 apresenta os experimentos realizados e por fim o Capítulo 5 apresenta a conclusão e os trabalhos futuros.

Classificação de padrões baseada em regras de decisão

2.1 Introdução

Ao contrário das regras de associação as regras de decisão têm os consequentes das regras pré-especificados e os mesmos nunca estão no antecedente de outra regra. As regras de decisão solucionam problemas de classificação pré-definidos, enquanto regras de associação normalmente são geradas a partir de uma base de dados transacional e representam transações. Embora haja esta diferença entre regras de decisão e regras de associação, a obtenção destas está fortemente relacionada. Sendo assim, é comum encontrarmos trabalhos que fazem uso de regras de associação para obtenção de regras de decisão.

As regras de associação têm a vantagem de fazer uma busca por todo tipo de relação existente no domínio do problema. Como os métodos de obtenção de regras de associação não seguem nenhuma heurística para confecção de regras, isto é, apenas a existência de relação entre atributos já é suficiente para uma nova regra existir, normalmente os classificadores baseados em regras de associação [63] têm uma alta precisão de classificação. A principal desvantagem dos classificadores que fazem uso de regras de associação é o alto número de regras geradas. Existem algumas propostas com objetivo de contornar esta questão, algumas destas são [53], [63], [59], [5], [60]. Tais propostas fazem uso de algum pós-processamento para eliminar estas regras redundantes ou desnecessárias obtidas de modo associativo. Este pós-processamento na maioria das vezes faz uso de heurísticas ou de algum algoritmos gulosos para refinamento da base de regras.

Os métodos para obtenção de regras de decisão podem fazer uso de alguma heurística. O uso de heurísticas para geração de regras pode fazer com que uma regra ótima não seja gerada, porém uma solução com uma quantidade menor de regras é possível ser obtida inicialmente, algumas propostas são [14], [13], [28], [63], [7], [20].

Uma regra de decisão if-then pode ser *crisp* se utiliza a lógica tradicional bi-valorada (verdadeiro ou falso) de Aristóteles [36]. Seja X o conjunto universo da aplicação. Seja μ_A e μ_B os graus de pertinência de um elemento $x \in X$ ao conjunto A e B , respectivamente. O conjunto A é dito *crisp* se $\mu_A : X \rightarrow \{0, 1\}$, ou seja, cada elemento de X pertence ou não a A .

B é um conjunto *fuzzy* [35] se $\mu_B : X \rightarrow [0, 1]$, ou seja, um elemento $x \in X$ pertence ao conjunto B com um grau de pertinência definido no intervalo $[0, 1]$. Podemos também ter regras *if-then fuzzy*, as quais utilizam a lógica *fuzzy* tri-valorada [35]. A teoria dos conjuntos *fuzzy* definida por Zadeh [35] trata a incerteza como um terceiro valor. Zadeh definiu um conjunto *fuzzy* como uma classe de objetos com um grau contínuo de pertinência.

Uma regra de decisão *if-then fuzzy* atribui um grau de pertinência de uma amostra desconhecida à classe predita pela regra. Como a incerteza com relação a pertinência de um objeto a uma classe é inerente em quase todos os problemas de classificação do mundo real, este trabalho tem como objetivo gerar regras de decisão *if-then fuzzy*. Este conjunto de regras será o núcleo de um sistema de classificação de padrões. A otimização de partições *fuzzy* [17] que representam esta incerteza vai além do escopo deste trabalho.

Na seção 2.2 são apresentados alguns trabalhos relacionados à classificação baseada em regras de decisão utilizando diferentes técnicas. Na seção 2.3 são apresentados os principais conceitos sobre a Teoria dos *Rough Sets* (TRS), necessários para o entendimento deste trabalho, assim como as propostas de classificação baseadas em regras de decisão que fazem uso da TRS.

2.2 Sistemas de classificação baseados em regras

Castro e Carmargo em [11] propuseram um processo genético para geração de regras *fuzzy if-then*. Numa primeira etapa os dados são pré-processados. As p funções de pertinência *fuzzy* para a partição *fuzzy* [20] (veja Figura 3.1 para exemplo de partição *fuzzy* composta por funções de pertinência *fuzzy*) para cada atributo são geradas utilizando o algoritmo de clusterização Fuzzy C-Means [61]. O Fuzzy C-Means é um conhecido algoritmo de clusterização que garante critérios de complementaridade e de completude. O critério de complementaridade indica que para cada elemento do universo de discurso a soma dos graus de pertinência para todos os conjuntos *fuzzy* deve ser igual a 1, isto garante distribuição de significado entre os elementos. A completude de sistemas *fuzzy* significa que para cada variável de entrada, pelo menos um conjunto *fuzzy* é atribuído. Os clusters formados definem a partição *fuzzy* de cada atributo.

Após geradas as partições *fuzzy* são aplicados dois AG's para obtenção da base final de regras. No primeiro AG é obtida uma base de regras como solução. O segundo AG otimiza esta base de regras.

O primeiro AG para geração da base de regras segue a abordagem de Pittsburg (cada indivíduo da população representa uma solução - base de regras). As regras são codificadas utilizando um índice para cada atributo e um índice para a classe. O índice com valor zero indica que o atributo não é utilizado ou *don't care*. A população inicial é gerada aleatoriamente. A função objetivo é o número máximo de padrões corretamente classificados pela base de regras codificada no indivíduo. Durante as operações genéticas a probabilidade de mutação é maior do que a probabilidade de *crossover*, para manter mais condições *don't care* na parte antecedente das regras. O critério de parada é o número máximo de iterações. A solução retornada é o indivíduo (base de regras) com melhor valor para a função objetivo.

Em seguida é aplicado um outro AG para eliminar regras redundantes e desnecessárias na solução retornada pelo AG descrito anteriormente. Neste segundo AG as operações genéticas

e o critério de parada são os mesmos do AG anterior. A codificação é feita de modo que cada regra da solução anterior seja representada por um *bit* de ativação. A população inicial é gerada pela introdução de um cromossomo que representa todas as regras obtidas previamente, isto é, todos os *bits* de ativação deste cromossomo tem valor 1 - ativo. Os demais cromossomos são gerados aleatoriamente. A função objetivo é agora baseada em duas medidas: número de amostras classificadas corretamente e número de regras ativas no cromossomo. Este segundo AG executa k iterações (ou evoluções). A solução final é o indivíduo mais adaptado (com melhor valor para a função objetivo).

Resultados experimentais foram realizados em 5 bases de dados públicas obtidas do UCI *Machine Learning Repository* [1]. Os resultados mostraram que a proposta gera bases de regras precisas quanto a classificação e compreensíveis quanto ao número de regras e ao tamanho das regras quando comparada com trabalhos anteriores.

Tan et. al. em [7] propuseram um algoritmo evolutivo com dois objetivos denominado *Dual-Objective Evolutionary Algorithm* (DOEA) para extração de regras de decisão. A proposta incorpora o conceito de *dominância de Pareto* [44] para obter um conjunto de regras não dominadas. Este conceito diz que uma solução x é considerada melhor do que uma solução y se ela é melhor ou igual a y em todos os objetivos e é melhor que y em pelo menos um objetivo. Nesta situação diz-se que x é uma solução não dominada [15], sendo denominada como solução ótima de Pareto [7]. O conjunto de todas as soluções não dominadas formam a *fronteira de Pareto* [7]. A proposta determina também um intervalo que limita o número de regras geradas.

A obtenção da base final de regras passa por duas fases. Na primeira fase é obtida, utilizando um processo evolutivo, uma base de regras candidatas para a segunda fase. Este processo evolutivo da primeira fase segue a abordagem de Michigan [62] em que cada indivíduo que compõe a população representa uma regra. Os indivíduos codificam apenas a parte antecedente de uma regra if-then, a parte consequente não precisa ser codificada porque o processo é executado separadamente para cada classe.

Na primeira fase a população inicial é gerada aleatoriamente. Cada indivíduo na população é avaliado sobre as amostras de treinamento segundo o valor de uma função objetivo. Em seguida são executadas operações genéticas de *crossover* e mutação antes de realizar uma competição por *tokens* (cada amostra de treinamento é um *token*). O critério de parada é o número de iterações.

Nesta primeira fase o indivíduo é representado por genes, em que cada gene é um atributo da base de dados. Todos os indivíduos têm o mesmo tamanho e o número de genes é igual ao número de atributos na base de dados. A um gene do indivíduo (futuro termo antecedente de uma regra) é atribuído um peso que determina seu uso ou não, segundo um limiar mínimo desejado.

Uma lista de 6 operadores relacionais (não há referência de onde esta lista foi retirada) é utilizada nas regras na comparação do valor dos atributos.

A função objetivo utilizada é composta por duas partes, a primeira parte é baseada em taxas de *true positive* (tp), *false positive* (fp), *true negative* (tn), *false negative* (fn) e os pesos w_1 e w_2 como mostrado na equação 2.1:

$$fo = \frac{tp + w1 * tn}{tp + w1 * tn + fp + fn} * \frac{1}{1 + w2 * fp}, \quad (2.1)$$

Note que é atribuído um peso $w1$ com objetivo de privilegiar indivíduos com maior taxa de tp . Quanto menor o valor de $w1$ maior ênfase é dada à medida de tp . O peso $w2$ é utilizado para punir indivíduos com alta taxa de fp .

A segunda parte da função objetivo é obtida utilizando a competição por *tokens*. Cada amostra na base de treinamento é um *token* pelo qual todos os indivíduos competem para capturar. Um indivíduo tem a oportunidade de capturar um *token* se todos seus antecedentes casam com o *token* e se a classe predita pelo indivíduo é a mesma do *token*, isto é, um caso de tp . Se mais de um cromossomo é eleito para capturar um *token*, vence o cromossomo com maior valor para fo . Após esta competição por *tokens* é computada a segunda parte da função objetivo e a função objetivo final para cada indivíduo, como definido na equação 2.2.

$$foa = fo * \frac{\text{número de tokens capturados}}{\text{número de tokens da classe}} \quad (2.2)$$

Após executadas um dado número de iterações, temos o final da primeira fase. São então selecionados os indivíduos (regras) que satisfazem um suporte mínimo [49]. Estas são a entrada para a segunda fase do AG.

Na segunda fase há um novo processo evolutivo. A população inicial deste processo é formada por um conjunto de regras obtido fazendo uma seleção e combinação aleatória dos indivíduos finais (conjuntos de regras) obtidos na primeira fase. Após obtenção destes conjuntos são realizadas operações genéticas de mutação e *crossover*. Para evitar regras idênticas e redundantes é realizada uma filtragem após as operações genéticas. Após esta filtragem é obtido o *rank de Pareto* no qual cada conjunto de regras não-dominadas é rankeado de acordo com a precisão de classificação do conjunto e o número de regras que compõe o conjunto. Os conjuntos não dominados de Pareto com k regras por conjunto são obtidos. Estes conjuntos são então combinados em uma população comum. A partir desta população comum são obtidas as regras não dominadas (*fronteira de Pareto*) como solução final.

A proposta foi avaliada em 8 bases de dados do UCI *Machine Learning Repository* [1] com atributos com valores nominais e numéricos. As bases foram divididas em 2/3 para treino e 1/3 para teste. Os resultados mostraram uma precisão de classificação próxima a de outros trabalhos existentes e mostraram que a base de regras é pequena.

Ishibuchi em [20] propôs um AG multiobjetivo para seleção de regras. O algoritmo é composto por 2 fases. Na primeira fase é feita a extração de regras candidatas. Na segunda fase é feita a seleção das regras que irão compor a base final de regras. Na primeira fase, as regras candidatas são geradas de modo aleatório. Para tanto, os termos linguísticos dos antecedentes das regras são determinados com base em 4 partições *fuzzy* fixas propostas em vários trabalhos anteriores [19], [21]. Cada termo linguístico de uma dada partição *fuzzy* mapeia o valor (estado) de um atributo. Todas as possíveis combinações de termos linguísticos podem compor a parte antecedente de uma regra *if-then*. A parte consequente das regras é determinada pela classe com a qual a parte antecedente tem maior compatibilidade. Se não existir nenhuma amostra de treino definida no subespaço da parte antecedente, a regra não é gerada. Uma

vez geradas as regras candidatas, são selecionadas, para compor a população inicial, apenas as regras que obedecem um número máximo permitido de termos antecedentes (pré-trata a questão da interpretabilidade) e que tem um grau mínimo de confiança e suporte.

Na segunda fase, o modelo de algoritmo *A fast and elitist multiobjective genetic algorithm* (NSGA-II) proposto por [12] é utilizado para obtenção da base final de regras. O aspecto especial do NSGA-II é o fato da utilização do *ranking de Pareto* na seleção dos pais para as operações genéticas e na atualização da população com os filhos gerados após estas operações.

Os três objetivos a serem otimizados durante a evolução do AG são: maximização do número de amostras corretamente classificadas, minimização do número de regras em cada solução (indivíduo) e minimização do número total de condições antecedentes (isto é, o tamanho total de todas as regras da solução). É mantido um número fixo de iterações, e a solução final é o melhor conjunto de indivíduos não dominados de Pareto (*fronteira de Pareto*). A classificação é feita de acordo com a regra que apresenta melhor casamento com a amostra desconhecida de teste. Os resultados experimentais foram realizados em duas bases de dados do UCI *Machine Learning Repository* [1]. Neste trabalho não foram apresentados em termos percentuais exatos a precisão de classificação. Contudo, são mostrados gráficos que deixam explícito o ganho na solução da relação entre obter regras compreensíveis e conseguir boa precisão de classificação.

Hong e Lee [58] propuseram um processo genético e um método de avaliação baseado na cobertura para geração de uma base de regras de decisão. A proposta consiste de três fases:

1. geração de regras *fuzzy* aleatoriamente,
2. codificação das regras *fuzzy*,
3. evolução das regras *fuzzy*.

Na primeira fase ocorre uma geração aleatória de N regras *fuzzy* (não é detalhado como é definido N). As partições *fuzzy* utilizadas na geração das regras têm número fixo m de funções de pertinência *fuzzy* e a forma destas funções de pertinência é dada. As regras têm tamanho variável, sendo geradas regras para cada uma das k classes. Na segunda fase, estas N regras serão codificadas para serem utilizadas no algoritmo genético da terceira fase. Cada regra é um indivíduo da população, e é codificada como uma string de *bits*. Cada atributo é representado por m *bits* (1 é ativo 0 inativo). É ativo o *bit* correspondente a função de pertinência utilizada, os demais *bits* ficam desativados. Todos os bits são ativos caso o atributo não seja utilizado. A classe é representada por k *bits* (k é o número de classes possíveis), fica ativo o *bit* correspondente a classe predita pela regra. Este método de codificação não utiliza espaço desnecessário ao utilizar m *bits* para representar cada atributo e k *bits* para representar uma classe, porque, na terceira fase esta representação é útil. Na terceira fase, o AG proposto seleciona os melhores indivíduos, de acordo com uma função objetivo, para gradualmente gerar regras (filhos) melhores.

Devido a forma como os indivíduos são codificados, após as operações genéticas de crossover *single point* (em apenas um ponto do indivíduo) e mutação, ocorre um processo de fissão (quebra do indivíduo para obter suas variações possíveis) para eliminar ambiguidade quanto a existência de mais de um *bit* ativo com relação aos m *bits* que representam o valor de um atributo ou com relação aos k *bits* que representam uma classe. Este processo de fissão substitui o indivíduo

que possui mais de um *bit* ativo para um dado atributo por todos os indivíduos possíveis (um indivíduo diferente para cada bit ativo).

Durante a evolução ocorre a maior contribuição deste trabalho, pois é apresentado um processo de medida de afinidade dos indivíduos e é feito uso de um conjunto de amostras de teste para cálculo do valor da função objetivo. A função objetivo utilizada durante o processo de medida de afinidade dos indivíduos leva em consideração a precisão, utilidade e cobertura com relação as amostras utilizadas como teste. A precisão de uma regra é o número de amostras corretamente classificadas. A utilidade de uma regra para as amostras desta base de teste é a soma da razão de 1 dividido pelo número de regras que classificam corretamente uma amostra de teste.

Em seguida é executado um processo de medida de afinidade que consiste em obter uma lista R_N com N indivíduos ordenados em ordem decrescente pelo resultado do produto entre a precisão e a utilidade. Ocorre então uma competição e cooperação entre os indivíduos para cálculo do valor da função objetivo. Neste processo, para o primeiro indivíduo R_1 (com maior produto entre precisão e utilidade) são definidas suas amostras cobertas na base de teste, para o segundo indivíduo R_2 também são definidas as amostras cobertas na base de teste e assim por diante. Se uma amostra casa melhor com R_2 do que com R_1 esta é transferida para o conjunto de amostras cobertas de R_2 . Esta estratégia coopera para a obtenção de um conjunto de regras concisas, pois regras com menos termos em sua parte antecedente tendem a ter maior cobertura. A função objetivo é então calculada como o produto da precisão, utilizada e cobertura (cardinalidade do conjunto de amostras cobertas).

A população sofre k iterações, até que uma base de regras *fuzzy* otimizada seja obtida. Após as k iterações as regras da população são combinadas de modo a compor o conjunto final de regras. Não são apresentados resultados experimentais, contudo a proposta em si é interessante.

2.3 Sistemas de classificação utilizando Teoria dos Rough Sets

A Teoria dos *Rough Sets* - TRS foi desenvolvida por Zdzislaw Pawlak [66] no início dos anos 1980. Esta teoria foi apresentada como um novo modelo matemático para representação do conhecimento, tratamento de vagueza (*vagness*) em dados para classificação.

Existem outras teorias como a teoria dos *fuzzy sets* [35], teoria de Bayes [42], teoria da evidência [26] empregadas no tratamento de conhecimento incerto. Em todas estas teorias se faz necessário informação *a priori* (função de pertinência, distribuição de probabilidade, crença, etc) a respeito dos dados. Na TRS não é necessário nenhuma informação *a priori*.

A TRS propõe um modelo de representação dos dados que é em si interessante, pois facilita a análise de informações existentes em uma base de dados. No contexto da construção de classificadores, a TRS pode ser aplicada para clusterização, seleção de características e geração de regras [67]. Na próxima subseção são apresentados os principais conceitos sobre esta teoria, essenciais para o entendimento deste trabalho. Na subseção 2.3.3 são abordados trabalhos que fazem uso da TRS no processo de geração de regras para a tarefa de classificação.

2.3.1 Conceitos básicos sobre Teoria dos *Rough Sets*

Seja uma base de dados representada como uma *tabela de dados*, na qual as linhas representam as amostras pertinentes ao domínio do problema e as colunas desta tabela de dados representam os atributos das amostras. Cada atributo pode ser uma propriedade, uma característica extraída ou qualquer outra informação observada e mensurada nas amostras. Esta tabela é também chamada de Sistema de Informação (SI), e formalmente é definida como a tupla (U, A) , em que U e A são conjuntos finitos não vazios, sendo U o conjunto de amostras chamado universo e A o conjunto de atributos presentes na *tabela de dados*. O valor de um atributo A para uma dada amostra x é definido como $x(a)$.

A classificação é muitas vezes conhecida para as amostras de uma *tabela de dados*, isto é, há um atributo que representa a classe a qual cada amostra pertence. Quando isto ocorre a *tabela de dados* passa a ser denominada uma *tabela de decisão* também chamada de Sistema de Decisão (SD). Para representar esta *tabela de decisão* formalmente, o conjunto A é particionado em dois subconjuntos C e D de atributos, e a *tabela de decisão* é definida pela tupla (U, C, D) , em que os atributos em C são chamados de *atributos condicionais* e os atributos em D são chamados *atributos de decisão* (classe), veja Figura 2.1

	U	C			D
	amostra	cor	tamanho	textura	classe
	x_1	Azul	Grande	Suave	X_1
	x_2	Vermelho	Médio	Suave	X_1
	x_3	Vermelho	Médio	Áspero	X_1
	x_4	Vermelho	Grande	Suave	X_1
	x_5	Azul	Grande	Suave	X_1
	x_6	Verde	Médio	Áspero	X_2
	x_7	Amarelo	Pequeno	Áspero	X_2
	x_8	Amarelo	Pequeno	Suave	X_2
	x_9	Vermelho	Grande	Suave	X_2
	x_{10}	Verde	Pequeno	Áspero	X_2

Figura 2.1: Exemplo de *tabela de decisão*

Relação de Indiscernibilidade

Todo o conhecimento disponível em uma base de dados está representado na *tabela de decisão*. Muitas vezes esse conhecimento é redundante, vago e/ou impreciso. O uso da TRS permite identificar estes aspectos. Para isto, o conceito de relação de indiscernibilidade (similaridade) entre as amostras de U é importante. A relação de indiscernibilidade é a base matemática para a TRS. Considere a *tabela de decisão* $= (U, C, D)$. Qualquer sub-conjunto $B \subseteq C$ determina uma relação binária $I(B)$ em U , denominada *relação de indiscernibilidade*, definida como:

$xI(B)y$ se e somente se $x(a) = y(a)$ para todo $a \in B$, sendo que $x(a)$ denota o valor do atributo a para o objeto x .

A relação $I(B)$ é uma relação de equivalência. Cada grupo de amostras indiscerníveis, obtido segundo a relação $I(B)$, é uma classe de equivalência. Formalmente uma classe de equivalência contendo uma amostra x é referenciada como $B(x)$. A família de todas as classes de equivalência de $I(B)$ é denotada por U/B . Cada uma das classes de equivalência em U/B , obtidas da relação $I(B)$, é denominada *conjunto elementar de B* . Para o sub-conjunto de atributos $B = \{\text{cor}, \text{tamanho}\}$ no exemplo de *tabela de decisão* apresentado na Figura 2.1 temos os seguintes conjuntos elementares : $U/B = \{\{x_1, x_5\}, \{x_2, x_3\}, \{x_4, x_9\}, \{x_6\}, \{x_7, x_8\}, \{x_{10}\}\}$. A Figura 2.2 ilustra graficamente U/B .



Figura 2.2: Conjuntos elementares para a *tabela de decisão* da Figura 2.1 , considerando $B = \{\text{cor}, \text{tamanho}\}$.

Aproximações

Os conjuntos elementares em U/B representam um particionamento do universo U utilizando o subconjunto de *atributos condicionais* B , segundo a relação de indiscernibilidade $I(B)$. Como um conjunto elementar é obtido a partir de U sem considerar o *atributo de decisão* D , pode ocorrer deste conter amostras iguais quanto ao espaço característica porém pertencentes a classes diferentes. Um conjunto elementar que contém amostras de classes diferentes apresenta um conhecimento impreciso sobre os dado, que pode ser representado por *Rough Sets*.

A TRS define duas aproximações, denominadas aproximação *lower* e aproximação *upper*, para a partir dos conjuntos elementares, em U/B , gerar novos conjuntos mais interessantes. Estes novos conjuntos gerados permitem um melhor entendimento do conhecimento disponível

na tabela de decisão, pois permitem diferenciar as amostras que são com certeza classificáveis das amostras que são possivelmente classificáveis ou não classificáveis.

Para formalizar estas duas aproximações de conjuntos, suponha um conjunto $X \subseteq U$. O conjunto X pode ser aproximado, usando as informações contidas em B , pelas aproximações *lower* e *upper* :

- aproximação *lower* representa o conjunto dos objetos que com certeza podem ser classificados como X utilizando B , definida como :

$$B_*(X) = \{x \in U | B(x) \subseteq X\}. \quad (2.3)$$

- aproximação *upper* representa o conjunto de todos os objetos que possivelmente podem ser classificados como X utilizando B , definida como :

$$B^*(X) = \{x \in U | B(x) \cap X \neq \emptyset\}. \quad (2.4)$$

Os dois conjuntos definidos pelas aproximações *lower* e *upper* determinam três regiões de interesse para um determinado conjunto X , denominadas região de fronteira, região positiva e região negativa. As Figuras 2.3 e 2.4 ilustram estas aproximações e regiões de interesse para as classes X_1 e X_2 , respectivamente, considerando $B = \{cor, tamanho\}$.

A aproximação *lower* é restritiva e gera um subconjunto de amostras que com certeza pertence a classe sendo analisada. A aproximação *upper* é permissiva e pode conter um subconjunto de amostras que não pertencem a classe sendo analisada. Portanto, com base nas características do subconjunto de amostras obtido pela aproximação *lower* é possível fazer uma afirmação totalmente segura sobre os dados, enquanto que a aproximação *upper* gera um subconjunto que não permite uma afirmação totalmente segura sobre os dados.

2.3.2 Função de pertinência *rough*

A um conjunto elementar $B(x)$ ou a um elemento x pode ser atribuído um grau de pertinência $\mu_X^B(x)$ que expressa o grau com que $B(x)$ ou x pode ser incluído ou pertencer a X respectivamente. O valor de $\mu_X^B(x)$ é definido como:

$$\mu_X^B(x) = \frac{|B(x) \cap X|}{|B(x)|}. \quad (2.5)$$

Na Figura 2.3 , para o conjunto elementar $\{X_4, X_9\}$, $\mu_X^B(x) = 0.5$ para a classe X_1 e $\mu_X^B(x) = 0.5$ para a classe X_2 , ou seja, 50% dos seus elementos pertencem à classe X_1 e 50% pertencem à classe X_2 . Já por exemplo, o conjunto elementar $\{X_7, X_8\}$ apresenta $\mu_X^B(x) = 0.0$ para a classe X_1 e $\mu_X^B(x) = 1.0$ para a classe X_2 . Neste caso, todas as amostras do conjunto elementar pertencem a classe X_2 .

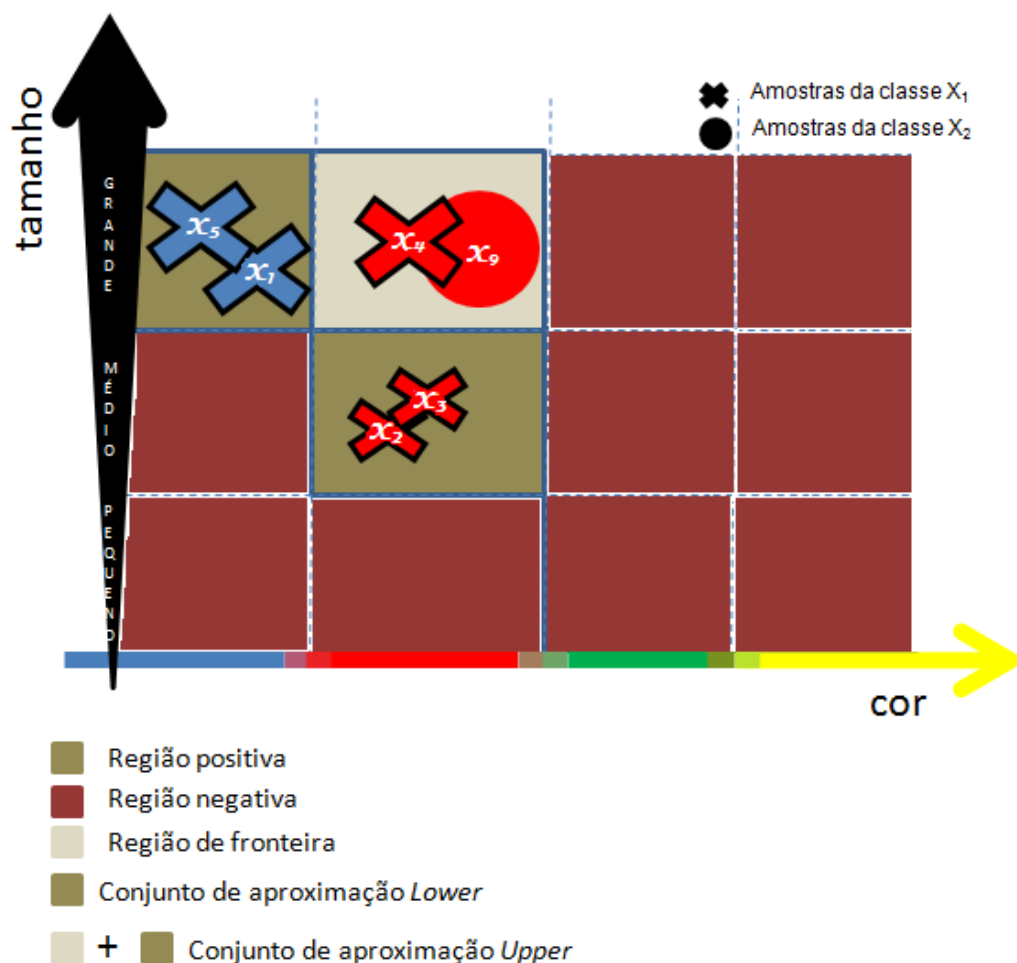


Figura 2.3: Conjuntos de aproximações *lower* e *upper* e regiões de interesse para a classe X_1 : $B_*(X_1) = \{x_1, x_2, x_3, x_5\}$; $B^*(X_1) = \{x_1, x_2, x_3, x_4, x_5, x_9\}$.

Redução de atributos

A redução de atributos [67], definida na TRS, não é empregada nesta dissertação, pois é computacionalmente cara e não ofereceu ganhos quando incorporada ao método proposto. Sua conceituação nesta subseção é apenas para entendimento de uma observação obtida ao realizar os resultados experimentais do método proposto.

Frequentemente lidamos com *tabelas de dados* com grande número de *atributos condicionais*, os quais muitas vezes são redundantes e/ou não trazem nenhuma informação relevante para o discernimento entre as amostras.

Espaço de características, com alta dimensionalidade, torna o sistema de classificação mais complexo, com maior tempo de treinamento e pode reduzir a capacidade de generalização do sistema [57]. Além disto, um grande número de características tende a gerar redundância de informação, o que pode prejudicar o desempenho de um sistema de classificação.

Esta queda no desempenho do classificador devido ao aumento da dimensionalidade do es-

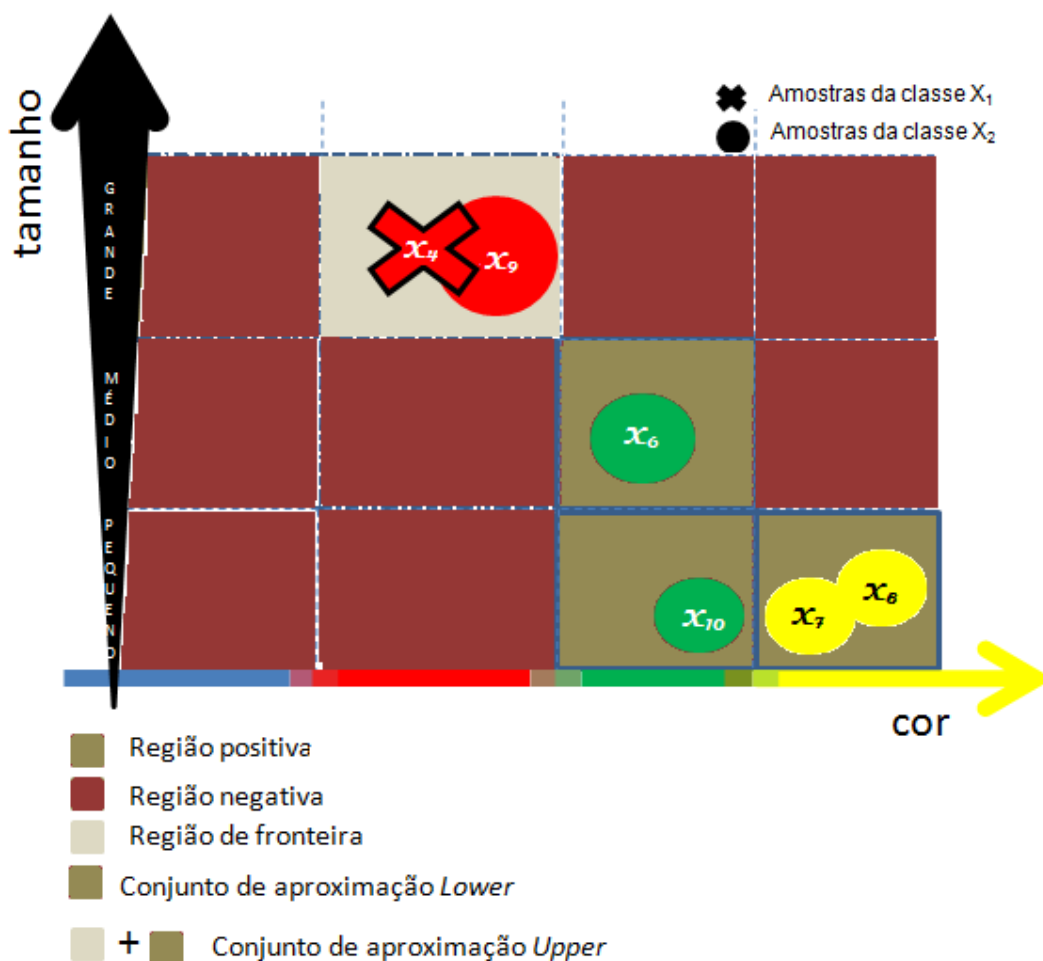


Figura 2.4: Conjuntos de aproximações *lower* e *upper* e regiões de interesse para a classe X_2 : $B_*(X_2) = \{x_6, x_7, x_8, x_{10}\}$; $B^*(X_2) = \{x_4, x_9, x_6, x_7, x_8, x_{10}\}$.

paço de características é um fenômeno conhecido como a maldição da dimensionalidade (*curse of dimensionality*) [56]. A maldição da dimensionalidade diz que dado um conjunto de treinamento (tabela de decisão), a classificação só melhora até um determinado número ótimo de dimensões, a partir do qual, o desempenho do classificador piora [32], [39]

Para reduzir o número de atributos Zdzisiaw Pawlak [67] propôs a obtenção de subconjuntos mínimos de *atributos condicionais* mantendo a mesma classificação para as amostras do universo U [67]. Deste modo, os atributos que não pertencem a um reduto são supérfluos. Um reduto é portanto um subconjunto de atributos com a menor cardinalidade (relativa ao número total de atributos) possível de ser obtida sem que seja alterada a partição e a consequente classificação das amostras pertencentes ao universo U .

Considere b um atributo de $B \subseteq C$, formalmente temos:

- Um atributo b pode ser dito supérfluo se $I(B) = I(B-b)$, caso contrário b é indispensável em B ,

- O conjunto B é *independente* se todos os seus atributos são indispensáveis,
- Um subconjunto B' de B é um reduto de B se B' é *independente* e se os conjuntos de amostras indiscerníveis são iguais para B' e B , ou seja, $I(B') = I(B)$.

2.3.3 Trabalhos relacionados

Cao et al. [64] desenvolveram um sistema baseado em regras de decisão para predição de proteínas. Os autores utilizaram a plataforma Rosetta [2] a qual implementa muitos algoritmos baseados na TRS. Na plataforma foi configurado o algoritmo *Semi Naive* para a discretização da base de dados numérica e algoritmo genético, baseado na teoria dos *rough Sets*, para a redução do espaço característica. Um método, não detalhado, para mineração da base de regras a partir do espaço característica reduzido foi aplicado. Neste trabalho a TRS foi utilizada apenas para redução do espaço característica.

Sarkar [40] propôs um método não paramétrico de classificação denominado FRNN - *Fuzzy-rough Nearest Neighbor Algorithm*. O método não inclui a letra K pois não necessita de informação acerca do número de vizinhos como no tradicional KNN - *K-Nearest Neighbor* ou no *fuzzy KNN* [56]. O método é capaz de distinguir entre a evidência e a ignorância no processo de classificação [17], [52].

Na Figura 2.5 é apresentado o algoritmo proposto. Como dados de entrada no algoritmo considere x_i o i -ésimo padrão de treinamento, $1 \leq i \leq n$ em que n é o número de padrões de treinamento, e y um padrão de teste. A variável $o(c)$ é o valor da *fuzzy-rough ownership*, a qual indica o grau de pertinência de y a classe c , indicando também a ignorância na classificação, veja equação 2.6.

$$o_c(y) = \frac{1}{|X|} \sum_{x \in X} \left[\mu_{C_c}(x) \exp \left(- \sum_{i=1}^N k_i (y_i - x_i)^{2/(q-1)} \right) \right], \quad (2.6)$$

em que $\mu_{C_c}(x)$ é o grau de pertinência do padrão x à classe c , determinado pela equação 2.7. A variável k indica a largura da banda da função de similaridade *fuzzy* $\tilde{\mu}_y(x) = \exp(-k||y - x||^{2/(q-1)})$ que compõe a equação 2.6. A largura de banda, determinada por k , indica o ponto no qual $\tilde{\mu}_y(x)$ tem valor 0.5 (acima deste grau de pertinência não é mais possível afirmar, de acordo com $\tilde{\mu}_y(x)$, que o padrão y pertence ou é similar com grau algum a x). O parâmetro k pode ter seu valor fixo, ou proporcional ao inverso da distância média entre todos os vizinhos e o padrão de teste y , isto é, $k = \frac{1}{(2/|X|)} = \sum_{x \in X} ||y - x||^{2/(q-1)}$.

$$\mu_{C_c}(x) = \begin{cases} 0.51 + 0.49n_j/k & \text{if } j = c, \\ 0.49n_j/k & \text{if } j \neq c, \end{cases} \quad (2.7)$$

em que n_j é o numero de vizinhos encontrados que levam a j -ésima classe.

Invés de utilizar um valor fixo para k ou adotar o proporcional ao inverso da distância média entre todos os vizinhos e o padrão de teste y , Sarkar tornou k sensível ao espaço característica. Para tanto, utilizou um vetor N-dimensional na forma de k , como definido na equação 2.8. O parâmetro $q \in (0, \infty)$ determina a forma da função de pertinência $\tilde{\mu}_y(x)$. Quando o valor de

q é próximo de 1 a função de pertinência tende a ser *crisp* com inclinação muito acentuada. Quando q tende ao infinito a inclinação é quase plana e a função de pertinência é maximamente *fuzzy*. O papel do parâmetro q é semelhante ao índice de *fuzziness* do algoritmo de clusterização C-Means [30].

$$k = [k_1, k_2, \dots, k_N]' = \left[\frac{|x|}{2 \sum_{x \in X} \|y_1 - x_1\|^{2/(q-1)}}, \frac{|x|}{2 \sum_{x \in X} \|y_2 - x_2\|^{2/(q-1)}}, \dots, \frac{|x|}{2 \sum_{x \in X} \|y_N - x_N\|^{2/(q-1)}} \right] \quad (2.8)$$

INPUT: (a) Training data $\{x_i \mid i = 1, 2, \dots, n\}$ with fuzzy class labels.
 (b) The test pattern y .

ALGORITHM:

Compute κ .
 FOR $c = 1$ to C
 Set $o(c)$ as zero
 END FOR
 FOR $i = 1$ to n
 Determine squared weighted distance
 $d = \sum_{j=1}^N \kappa_j (y_j - x_{ij})^2$ between y and x_i .
 FOR $c = 1$ to C
 L: $o(c) = o(c) + \frac{1}{|\mathcal{X}|} \mu_{c_c}(x_i) \exp(-d^{1/(q-1)})$
 END FOR
 END FOR

Crisp class label of y is j where $o(j) = \max\{o(1), o(2), \dots, o(C)\}$.

OUTPUT: (a) Class label of y .
 (b) Class confidence values $o(c) \forall c$.

Figura 2.5: Algoritmo proposto por Sarkar.

Após o cálculo de k , o valor de $o(c)$ para as C classes pertencentes ao domínio do problema são setadas como 0 (zero). Para cada uma das n amostras é calculada a distância ponderada, segundo o valor de k , para as N características e armazenado em d . L armazena a soma acumulada da *fuzzy-rough ownership* da amostra de teste y e cada um dos n padrões x_i . O padrão de teste y é então classificado como sendo da classe j que obtiver o maior valor acumulado em L , $o(j) = \max\{o(1), o(2), \dots, o(C)\}$, e cada classe C tem grau de confiança, $o(c) \forall C$. Se a soma do grau de confiança para todas as classes é quase nulo há uma total ignorância durante a classificação.

Foram realizados 3 (três) experimentos, o primeiro para classificar as vogais 'a', 'e', 'i', 'o', 'u' cuja base de dados está disponível em <http://www.geocities.com/fuzzy.rough/speechData.ht>. Os resultados foram superiores, em termos de precisão de classificação, quando comparados com classificador de Bayes, rede neural *back propagation*, KNN e *Fuzzy KNN* [56]. Para o segundo experimento foi utilizada a base de dados *letter image recognition* obtida no UCI Machine Learning Repository [1]. Desta base foram utilizadas as letras 'D', 'G', 'O', 'Q'. Os resultados, em termos de precisão de classificação, foram comparados com o KNN e o *Fuzzy KNN* e se

mostraram superiores. O terceiro experimento teve como objeto de avaliação uma base de dados para diagnosticar se o paciente tem ou não sérios ferimentos na cabeça. Os resultados também se mostraram superiores quando comparados com o KNN e o *Fuzzy KNN*.

Hong et al. [46] integraram as teorias de *fuzzy sets* e *rough sets* (*fuzzy rough sets*) para produzir todas as possíveis regras para uma base de dados numérica, levando em consideração uma relação de indiscernibilidade *fuzzy* entre os objetos da base de dados. Em [45], Hong et al. melhoram o método anteriormente proposto, de tal forma a gerar um menor número de regras com a máxima cobertura possível. Inicialmente, cada valor numérico é transformado em um termo linguístico usando funções de pertinência *fuzzy*. Em seguida as aproximações *fuzzy lower* e *fuzzy upper* são calculadas. As regras *fuzzy* são obtidas a partir destas aproximações por um processo de indução iterativo.

Hong et. al em [23] propuseram um outro método para geração de regras de decisão baseado em uma precisão para os *rough sets*. Primeiro os atributos numéricos são transformados em termos linguísticos segundo partições *fuzzy* com funções de pertinência *fuzzy* pré-definidas. Depois são calculadas as aproximações $\beta - lower$ e $\beta - upper$ a partir das quais são obtidas regras que classificam com certeza e que possivelmente classificam são geradas a partir destas aproximações, respectivamente. Para tanto, é utilizada uma relação de indiscernibilidade *fuzzy*. Nesta relação, as amostras de um conjunto elementar são indiscerníveis se elas apresentam as mesmas funções de pertinência *fuzzy* (representadas por termos linguísticos) para cada atributo de um dado subconjunto B de atributos. Uma amostra y de um conjunto elementar tem grau de pertinência $\mu_{B_k}(y)$ ao conjunto elementar B_k , para tanto o menor valor de pertinência *fuzzy* entre os atributos em B é utilizado na relação de indiscernibilidade *fuzzy*.

Um conjunto elementar B_k tem um grau de erro de classificação com relação a um conjunto de amostras da classe X . Este grau de erro de classificação é baseado na razão entre o somatório do grau de pertinência $\mu_{B_k}(y)$ do conjunto elementar B_k que são da classe X e somatório do grau de pertinência $\mu_{B_k}(y)$ de todas as amostras do conjunto elementar B_K . Se todas as amostras do conjunto elementar pertencerem 100% ao conjunto X , o erro de classificação será 0, veja equação 2.9.

$$c(B_k(x), X) = 1 - \frac{\sum_{y \in (B_k(x) \cap X)} \mu_{B_k}(y)}{\sum_{y \in B_k(x)} \mu_{B_k}(y)} \quad (2.9)$$

A partir destes conjuntos elementares *fuzzy* são geradas os conjuntos para as aproximações $\beta - lower$ e $\beta - upper$. A aproximação $\beta - lower$ é composta pelas amostras de conjuntos elementares que tem grau de erro de classificação menor ou igual a um limiar β , veja equação 2.10 :

$$B_{*\beta}(X) = \{(B_k(x), \mu_{B_k}(x)) | x \in U, c(B_k(x), X) \leq \beta, 1 \leq k \leq |B(x)|\} \quad (2.10)$$

A aproximação $\beta - upper$ é composta pelas amostras de conjuntos elementares que tem grau de erro de classificação maior que um limiar β e menor que $1 - \beta$, veja equação 2.10 :

$$B_{\beta}^*(X) = \{(B_k(x), \mu_{B_k}(x)) | x \in U, \beta < c(B_k(x), X) < 1 - \beta, 1 \leq k \leq |B(x)|\} \quad (2.11)$$

Elementos em $B_{*\beta}(X)$ podem ser classificados como membros de X com um grau de erro de classificação β , já os elementos em $B_{\beta}^*(X)$ podem ser classificados com um grau de erro de classificação $1 - \beta$.

Para cada combinação de atributos B são obtidos os conjuntos de amostras gerados utilizando $B_{*\beta}(X)$ e $B_{\beta}^*(X)$.

A partir destas duas aproximações propostas são derivados dois conjuntos de regras de classificação para as amostras da classe X . O conjunto de regras $\beta - certeza$ que classificam com certeza e outro conjunto de regras que possivelmente classificam $\beta - possivel$, a partir de $B_{\beta}^*(X)$ e $B_{*\beta}(X)$, respectivamente.

Após a geração do conjunto de regras $\beta - certeza$ são removidas, deste conjunto de regras, aquelas que são mais específicas e que a medida de eficácia ou plausibilidade (mensurada como: $1 - c(B_k(x), X)$) é menor ou igual a alguma regra em $\beta - certeza$.

Do mesmo modo, após a geração de todas as regras $\beta - possivel$ são removidas, deste conjunto de regras, aquelas que são mais específicas e que a medida de efetividade ou plausibilidade é menor ou igual a alguma regra em $\beta - certeza$ ou $\beta - possivel$.

Este processo de derivação de conjuntos de regras é realizado para cada classe pertencente ao domínio do problema. A classificação é dada pela regra vencedora (aquela que melhor casa com a amostra desconhecida), caso a regra vencedora seja de $\beta - certeza$ a amostra desconhecida é classificada por uma classificação com certeza de pertencer à classe predita pela regra vencedora. Caso a regra vencedora seja de $\beta - possivel$ a amostra desconhecida é classificada por uma classificação parcial à classe predita pela regra vencedora.

Shen e Chouchoulas [48], propuseram uma técnica que integra um algoritmo de indução para geração de regras *fuzzy* com as aproximações da teoria dos *rough sets* para obter a redução do espaço característica. As regras são geradas por um algoritmo de indução de regras *fuzzy* exaustivo. A teoria dos *rough sets* é utilizada apenas na etapa de redução do espaço característica e não na geração de regras.

2.4 Discussão sobre os trabalhos relacionados

A proposta apresentada por Castro e Camargo em [11] faz uso de dois AG's e gerou bases de regras precisas e compreensíveis para as bases de dados utilizadas na validação da proposta. O primeiro AG tem como objetivo obter uma base de regras com alta precisão de classificação e com poucos termos antecedentes. No segundo AG esta base de regras é refinada fazendo uso de dois objetivos (maximização do número de amostras classificadas corretamente e minimização do número de regras ativas no cromossomo). O uso destes objetivos (heurísticas) pode levar a não geração de uma solução menos subótima do que alguma outra possível. Outra questão é o uso de dois AG's que requerem um grande número de parâmetros como taxa de mutação,

taxa de crossover, número de evoluções, tamanho da população inicial, entre outros, definidos empiricamente.

O algoritmo genético multiobjetivo apresentado por Tan et. al. em [6] gera regras precisas e compreensíveis. entretanto, o tamanho da base de regras depende da restrição com relação ao parametro n . Há um excesso de parametros (mais de 10) a serem calibrados na proposta. Para a base de dados Iris, na qual os atributos são todos numéricos, a proposta é menos competitiva quando comparada com outros trabalhos. Nas demais bases de dados, os resultados comparativos com propostas existentes até a data de realização do trabalho foram satisfatórios. Ishibuchi em [20] também apresentam um AG multiobjetivo que busca obter um ponto de equilíbrio entre o *tradeoff* entre a precisão e interpretabilidade da base de regras obtida.

Hong e Lee [58] apresentaram um método genético que durante a evolução obviamente faz uso de uma base de treinamento e a cada evolução utiliza uma outra base diferente da de treinamento para avaliar os indivíduos atuais. O modo como os indivíduos são avaliados é interessante pois ocorre uma competição por *tokens* (amostras). Quanto mais *tokens* um indivíduo (regra) conquista (cobre) maior é a sua cobertura. O trabalho não apresenta resultados experimentais mas a idéia proposta é atraente.

Sarkar [40] propôs um método de classificação denominado *Fuzzy-Rough Nearest Neighbor Algorithm*, capaz de distinguir entre a evidência e a ignorância no processo de classificação de um dado objeto. Embora esse classificador apresente semelhanças, do ponto de vista funcional, ao classificador aqui proposto, a abordagem utilizada é bastante diferente. Sarkar propõe um algoritmo que basicamente verifica, utilizando todas as características, o grau de pertinência da amostra de teste as demais amostras de treinamento. Para cada classe pertencente ao domínio do problema há uma conjunto de amostras de treinamento. A amostra de teste é na classe definida pelo conjunto de amostras da base de treinamento com a qual tem maior grau de pertinência acumulado para com as amostras que compõem o conjunto que a classifica. A ignorância no processo de classificação ocorre quando a amostra não pertence a nenhum conjunto com um grau de pertinência maior que 0.5, considerando o intervalo de pertinência contínuo [0,1]. Os resultados experimentais apresentam bons resultados para a precisão de classificação, entretanto o autor não apresenta o número de amostras que não foram classificadas em uma das classes definidas no domínio do problema pela falta de evidências (ignorância), e que não introduzem erro à precisão final. Um conhecimento sobre este número de amostras identificadas pela ignorância é relevante para análise dos resultados.

Hong et. al em [23] fazem uso de uma relação de indiscernibilidade *fuzzy* para obter um segundo conjunto de regras que, para um conjunto de amostras obtido da aproximação β -lower, classificam com certeza, e para obter um outro conjunto de regras que, para um conjunto de amostras obtido da aproximação β -upper, possivelmente classificam. Dividir as regras obtidas em dois conjuntos de regras não é relevante visto que de todo modo a amostra é classificada pela regra vencedora (aquela que a amostra desconhecida tem maior grau de pertinência) independentemente do conjunto a qual a regra pertence. O conjunto de regras que classificam com certeza apenas classificam segundo um limiar mínimo respeitado para o erro de classificação, enquanto que o conjunto de regras que possivelmente classificam, apenas classificam segundo um intervalo para o erro de classificação (composto inferiormente por um limiar mínimo e superiormente por um limiar imediatamente inferior ao limiar das regras que classificam com certeza). A partir do momento que é possível apenas um conjunto com todas as regras (regras

que possivelmente classificam e as que certamente classificam) dizer o grau de pertinência da amostra desconhecida a regra vencedora que a classificou, não se faz mais interessante dizer que uma amostra foi possivelmente classificada ou com certeza classificada segundo um limiar para este grau.

2.5 Considerações finais do capítulo

Neste capítulo foi apresentado os conceitos básicos da TRS e os trabalhos relacionados ao trabalho sendo proposto.

No próximo capítulo é apresentada a proposta para geração automática de regras *fuzzy* do tipo if-then deste trabalho e uma extensão (nova aproximação) à TRS que servirá como base para o método de geração de regras de decisão aqui proposto. A nova aproximação apresenta vantagens com relação as aproximações *lower* e *upper* e será utilizada como base para o processo de geração de regras.

O conceito sobre partição *fuzzy* é introduzido no momento da categorização dos valores dos atributos da base de dados, entretando, esta categorização não é o foco deste trabalho. O método de geração de regras proposto não exige um conhecimento prévio acerca da base de dados e produz um conjunto reduzido de regras concisas e com alta taxa de precisão de classificação. O conjunto de regras compõe o núcleo de um classificador que é capaz de distinguir entre a evidência e a ignorância no processo de classificação.

Proposta de um método para geração automática de regras *fuzzy* do tipo if-then

3.1 Introdução

Neste capítulo é apresentada a proposta de gerar regras *fuzzy* do tipo if-then. É descrito como foram realizadas cada uma das cinco etapas de um processo de KDD (veja seção 1.2). A etapa 1 (Pré-processamento) não é o foco deste trabalho e é descrita apenas para conhecimento de como foi realizada. O foco está principalmente na etapa 2 (DM) e na etapa 3 (Avaliação e interpretação do conhecimento extraído).

Dentro da etapa de DM é proposto um novo método supervisionado para geração de regras de decisão *fuzzy* if-then baseado em uma extensão proposta a TRS. Para tanto, o conhecimento disponível em uma tabela de decisão é explorado em diferentes dimensionalidades com diferentes combinações. Tais subconjuntos do conhecimento são referenciados neste trabalho como *grânulos do conhecimento*. O uso de diversos grânulos do conhecimento permite obter várias partições do universo do discurso, o que é muito importante para que seja possível identificar, utilizando o menor número de atributos possíveis, o conhecimento na base de dados relevante para classificação. Portanto, o método de geração de regras proposto pode dar prioridade a inclusão de regras mais concisas e com maior cobertura na base de regras do classificador.

Diante do conhecimento disponível em uma base de dados, uma dada amostra pode ser classificada ou não. A impossibilidade de classificação de uma amostra ocorre quando há igualdade de evidências ou ignorância durante o processo de classificação. A igualdade de evidências é identificada nas situações em que a amostra pertence a duas ou mais classes com o mesmo grau de evidência e a ignorância é identificada nas situações que o conhecimento disponível não apresentar qualquer evidência sobre a pertinência desta amostra a qualquer uma das classes da aplicação. As regras obtidas pelo método proposto, compõem o núcleo de um sistema de classificação possibilístico, capaz de identificar a evidência e a distingui-la da igualdade de evidências e da ignorância no processo de classificação[3]. A proposta foi denominada de *Classification Based on Rules using Upper^{alpha} - CBRU^α*.

3.2 Proposta de uma nova aproximação: $upper^\alpha$ de X

Esta seção apresenta a extensão proposta às aproximações *lower* e *upper* definidas por Pawlak [66] apresentadas na subseção 2.3.1.

Como pode ser notado, a aproximação *lower* de X é bastante restritiva e um conjunto gerado por ela é totalmente preciso, contendo apenas elementos cujos conjuntos elementares possuem $\mu_X^B(x) = 1$. A partir desta aproximação é possível identificar os conjuntos elementares a partir dos quais pode-se gerar regras if-then com alta precisão. As desvantagens da utilização desta aproximação na obtenção de regras de decisão são:

- regras com conjuntos elementares com baixa cardinalidade;
- número elevado de regras : utilização de muitos conjuntos elementares de pequena cardinalidade;
- baixa precisão na classificação de amostras desconhecidas : amostras que poderiam ser classificadas com um grau de incerteza aceitável são descartadas no treinamento;

Em contrapartida, a aproximação *upper* de X é bastante permissiva e o conjunto gerado pode ser altamente impreciso, pois pode conter elementos cujos conjuntos elementares com $0 \leq \mu_X^B(x) \leq 1$. Esta aproximação pode gerar por exemplo um conjunto contendo 50% de amostras de uma classe e 50% de amostras de outra classe, o que o torna pouco interessante para um processo de geração de regras, pois não traz discernimento algum entre as classes. Esta é a principal desvantagem desta aproximação.

Devido aos motivos acima relacionados, nenhuma das duas aproximações tradicionais definidas na TRS são apropriadas para serem utilizadas unicamente no processo de geração de regras. Para transpor este problema, este trabalho propõe uma extensão para estas duas aproximações definidas na TRS utilizando o conceito α -cut [17], $0 < \alpha \leq 1$: $B^{*\alpha}$ dedinido como:

$$B^{*\alpha}(X) = \{x \in U | B(x) \cap X \neq \emptyset, \mu_X^B(x) \geq \alpha\}. \quad (3.1)$$

A aproximação $B^{*\alpha}(X)$ contém todos os conjuntos elementares de U/B tal que $\mu_X^B(x) \geq \alpha$. O conjunto $B^{*\alpha}$ será utilizado para a geração automática de regras *fuzzy*, conforme descrito na subseção 3.3.2.

3.3 Proposta de um novo método para geração de regras fuzzy utilizando $upper^\alpha$

Dentro das 3 etapas que envolvem um processo de KDD a etapas 1 (Pré-processamento) é apresentada na subseção 3.3.1. A etapa 2 (DM), foco principal deste trabalho, é apresentada na subseção 3.3.2 e a etapa 3 (Avaliação e interpretação do conhecimento extraído) é apresentada no próximo capítulo.

3.3.1 Pré-processamento

Nesta etapa a seleção de amostras não é realizada, pois já estamos lidando com um domínio de problema específico representado em uma tabela de dados. O tratamento de valor perdido em uma amostra para um dado *atributo condicional* é preenchido com a média aritmética dos valores do respectivo atributo condicional.

As amostras que tenham atributos vazios não são descartadas por serem interessantes dentro da proposta deste trabalho, pois é feito uso de computação granular, deste modo, grânulos do conhecimento com atributos não vazios podem ser interessantes para o processo de KDD. Tais amostras, com atributos vazios, tem seus valores preenchidos com a respectiva média aritmética para as amostras da base.

Com relação a transformação dos dados não é aplicado nenhuma redução de dimensionalidade via seleção ou transformação de atributos [56]. Nesta etapa é realizada uma categorização *fuzzy* dos dados. A categorização de atributos numéricos pode causar perda de informação devido a partição utilizada [65], [50].

A partição *fuzzy* utilizada para categorizar os valores numéricos dos *atributos condicionais* normalizados para o intervalo $[0,1]$, é obtida pela divisão homogênea de cada dimensão do espaço de características em p termos linguísticos [17]. Neste trabalho, as funções de pertinência que representam os termos linguísticos têm a forma trapezoidal. O valor numérico, que representa um *atributo condicional*, é associado ao termo linguístico cuja função de pertinência *fuzzy* apresenta maior valor de pertinência. A base maior e a base menor do trapézio são determinadas pela inclinação dos outros dois lados e pelo valor de p . Neste trabalho estes dois lados tem tamanho de projeção sobre o eixo das coordenadas de $ic = 0.01$. O cálculo do tamanho da base menor é portanto: $b = (1 - ((p + 1) * ic))/p$. Com o valor de ic e da base menor b pode-se montar a partição *fuzzy*. Veja a partição mostrada na Figura 3.1, obtida com $p = 4$, $ic = 0.01$ e portanto $b = 0.2375$. Quanto maior o valor de p menor o comprimento da base menor b .

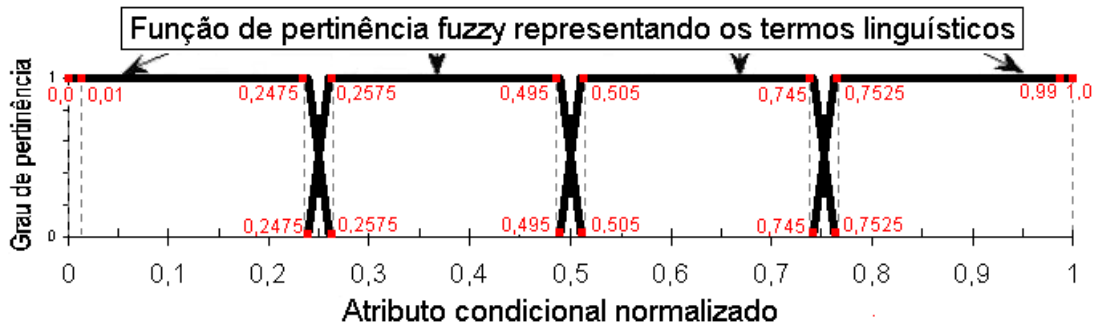


Figura 3.1: Partição *fuzzy* com $p = 4$.

Note que a escolha da partição *fuzzy* influencia diretamente na análise de indiscernibilidade dos objetos. Diferentes partições geram diferentes *conjuntos elementares*, gerando, em consequência, diferentes conjuntos de regras. A geração automática e refinamento de partições do espaço de características vai além do escopo deste trabalho.

3.3.2 Obtenção de regras de decisão fuzzy - DM

Definição dos grânulos do conhecimento

A exploração simultânea de diferentes grânulos do conhecimento, de uma mesma base de dados, no processo de reconhecimento de padrões permite identificar quais subconjuntos do conhecimento são mais adequados para representar cada agrupamento de dados ou amostras. Uma decorrência desta abordagem é a redução do espaço de características, seja pela remoção de atributos redundantes ou pela remoção de atributos com informações ambíguas.

Uma abordagem, com custo computacional baixo, para obtenção de diferentes grânulos do conhecimento, é o uso de AG's [41]. Foram implementados AG's durante o desenvolvimento deste trabalho na tentativa de obter os grânulos do conhecimento. Entretanto, devido a aleatoriedade no processo de geração e evolução de indivíduos e devido ao uso de função objetivo empírica, os AGs não garantem que todos os grânulos significativos estejam presentes no resultado final, principalmente em bases de dados com um número maior de *atributos condicionais*. Decidiu-se então, neste trabalho utilizar técnicas de mineração dos dados [47] para geração dos grânulos do conhecimento.

Grânulos do conhecimento podem ser obtidos a partir de *itemsets* presentes na base de dados. Considere um item um termo linguístico para um dado *atributo condicional* como um item. Um *itemset*, I_i , é uma dada combinação de itens. Uma das técnicas mais atraentes para tratar do problema de mineração de *itemsets* frequentes é o algoritmo Apriori [49]. Ele pode trabalhar com um número grande de atributos, gerando várias alternativas combinatórias entre eles. A principal propriedade deste algoritmo diz que dados dois *itemsets* $I1$ e $I2$ tal que $I1 \subset I2$, se $I2$ é frequente então $I1$ também é frequente. Deste modo, se um *itemset* é frequente todos os sub-*itemsets* contidos nele são frequentes. Um *itemset* que contém pelo menos um sub-*itemset* não frequente é descartado, e não está no conjunto final de *itemsets* e nem é sub-*itemset* de outro *itemset* resultante do Apriori.

Considere a tabela de decisão apresentada na Tabela 3.1 como exemplo para entendimento do funcionamento do Apriori. O algoritmo Apriori foi configurado de modo que sua saída é um conjunto de *itemsets* frequentes (considerando os atributos em C e em D). Veja o Apriori em Algoritmo 1.

Algoritmo 1 Algoritmo Apriori

ENTRADA : TD (tabela de decisão), S (suporte mínimo)

SAÍDA : Conjunto de *Itemsets*($F_1, F_2, \dots, F_{K-1}$)

$C_1 = \text{Itemsets de tamanho } 1$

$F_1 = \text{Itemsets frequentes de } C_1$

$K = 1$

while F_K não for vazio **do**

$C_{K+1} = \text{gera_unindo}(F_K, F_K)$

$C_{K+1} = \text{poda}(C_K, F_K)$

$F_{K+1} = \text{valida}(TD, C_{K+1}, S)$

$K = K + 1$

end while

Considere um suporte mínimo S de 0.5 (50%). Neste contexto, o conjunto de *itemsets* de tamanho 1 é $C_1 = \{\text{redondo}\}, \{\text{quadrado}\}, \{\text{leve}\}, \{\text{médio}\}, \{\text{pesado}\}, \{\text{média}\}, \{\text{alta}\}, \{\text{baixa}\}, \{d_1\}, \{d_2\}$; e os *itemsets* frequentes em C_1 é o conjunto $F_1 = \{\text{redondo}\}, \{\text{médio}\}, \{\text{alta}\}, \{d_1\}, \{d_2\}$. O algoritmo consiste de 3 etapas: geração, poda e validação. No laço **while**, durante a etapa de geração de *itemsets*, $\text{gera_unindo}(F_1, F_1)$, são unidos os *itemsets* em F_K que contém sub-*itemsets* de tamanho $K-1$ iguais (observação : para a primeira iteração, $K=1$, todos são unidos). O resultado da geração é armazenado em $C_2 = \{\text{redondo, médio}\}, \{\text{redondo, alta}\}, \{\text{redondo, } d_1\}, \{\text{redondo, } d_2\}, \{\text{médio, alta}\}, \{\text{médio, } d_1\}, \{\text{médio, } d_2\}, \{\text{alta, } d_1\}, \{\text{alta, } d_2\}, \{d_1, d_2\}$. A etapa de poda, $\text{poda}(C_2, F_1)$, remove de C_2 os *itemsets* que contém pelo menos um sub-*itemset* não existente em F_1 , seguindo a principal propriedade do Apriori apresentada anteriormente. Deste modo, C_2 é mantido, pois os sub-*itemsets* $\{\text{redondo}\}, \{\text{médio}\}, \{\text{alta}\}, \{d_1\}, \{d_2\}$ existem em F_1 . A etapa de validação considera a partir de C_2 apenas os *itemsets* que satisfizeram o suporte mínimo com relação a Tabela 3.1. Com a validação, tem-se $F_2 = \{\text{redondo, médio}\}, \{\text{redondo, alta}\}, \{\text{médio, alta}\}$. Fazendo do mesmo modo para $K=2$ tem-se na geração $C_3 = \{\text{redondo, médio, alta}\}$, na poda $C_3 = \{\text{redondo, médio, alta}\}$ e na validação $F_3 = \{\text{redondo, médio, alta}\}$. Para $K=3$ tem-se $C_4 = \{\}$ e $F_4 = \{\}$ e o algoritmo para. O resultado do Apriori é o conjunto de *itemset* frequentes $RA = F_1 \cup F_2 \cup F_3 = \{\text{redondo}\}, \{\text{médio}\}, \{\text{alta}\}, \{d_1\}, \{d_2\}, \{\text{redondo, médio}\}, \{\text{redondo, alta}\}, \{\text{médio, alta}\}, \{\text{redondo, médio, alta}\}$.

Tabela 3.1: Exemplo de tabela de decisão

x_i	Forma	Peso	Frequência	d_i
x_1	redondo	leve	média	d_1
x_2	redondo	médio	alta	d_1
x_3	redondo	médio	alta	d_1
x_4	quadrado	leve	baixa	d_1
x_5	quadrado	leve	baixa	d_1
x_6	redondo	médio	alta	d_1
x_7	redondo	leve	baixa	d_2
x_8	redondo	médio	alta	d_2
x_9	redondo	pesado	baixa	d_2
x_{10}	redondo	médio	alta	d_2
x_{11}	redondo	médio	alta	d_2
x_{12}	quadrado	pesado	alta	d_2

A partir de RA é agora executado um pós-processamento para obter os grânulos do conhecimento. Primeiramente é feita uma filtragem em RA de modo a eliminar os *itemsets* compostos apenas pelo valor de um *atributo de decisão* em $D = \{d_1, d_2\}$. Tem-se então agora $RA = \{\text{redondo}\}, \{\text{médio}\}, \{\text{alta}\}, \{\text{redondo, médio}\}, \{\text{redondo, alta}\}, \{\text{médio, alta}\}, \{\text{redondo, médio, alta}\}$.

Cada item em cada *itemset* em RA está associado a um atributo em C , por exemplo o *itemset* $\{\text{médio, alta}\}$ está associado aos atributos Peso e Frequência, respectivamente, portanto, $\{\text{Peso, Frequência}\}$ é um escolhido como grânulo do conhecimento. A partir de RA são então obtidos os W distintos grânulos do conhecimento B_g , $1 \leq g \leq W$. Para o exemplo temos $B_1 = \{\text{Forma}\}$, $B_2 = \{\text{Peso}\}$, $B_3 = \{\text{Frequência}\}$, $B_4 = \{\text{Forma, Peso}\}$, $B_5 = \{\text{Forma, Frequência}\}$, $B_6 = \{\text{Peso, Frequência}\}$, $B_7 = \{\text{Forma, Peso, Frequência}\}$. Note que para o exemplo dado, com suporte mínimo 0.5, todas as combinações possíveis de atributos são grânulos do conhecimento.

Método proposto para geração da regras de decisão fuzzy a partir de *upper*^α

A forma geral de uma regra fuzzy do tipo if-then é dada por:

R_r : IF a_1 is A_j^i and ...and a_n is A_j THEN $classe = d_i$,

sendo A_j^i o termo linguístico j associado ao atributo a_i , $1 \leq j \leq p_k$, e $1 \leq i \leq n$, em que p_k é o número de termos linguísticos que representam a k -ésima partição fuzzy e n é o número de termos antecedentes na regra.

Considere uma aplicação com M classes, n atributos, W grânulos do conhecimento obtidos pelo método descrito na seção anterior.

Seja o conjunto, $L_{d_i} = \bigcup_{g=1}^W U/B_g$. Cada elemento de L_{d_i} é uma tupla (CE_t, B_g) , em que CE_t é um conjunto elementar com grau de pertinência $\mu_X^B(x) \geq \alpha$ para a classe d_i . Durante a união, caso duas tuplas sejam idênticas com relação as amostras do conjunto elementar CE_t , é escolhido o conjunto elementar associado ao B_g com menor cardinalidade.

Cada tupla (CE_t, B_g) em L_{d_i} representa potencialmente uma regra. Para garantir regras com poucos termos antecedentes, os elementos de L_{d_i} são ordenados de acordo com a cardinalidade de B_g , de tal forma a priorizar o B_g com menor número de atributos. Em caso de empate, é prioritário o conjunto elementar CE_t que cobre mais amostras da classe d_i .

Definição de $B^\alpha(d_i)$

$B^\alpha(d_i) = \bigcup_{g=1}^W B_g^{\alpha}(d_i)$ é o conjunto que contém todos os objetos que podem ser classificados na classe d_i , considerando o valor de α e os diferentes grânulos do conhecimento B_g .

O conjunto de regras para cada classe d_i é obtido pela análise simultânea dos elementos de $B^\alpha(d_i)$ e de L_{d_i} . Cada par (CE_t, B_g) em L_{d_i} produz uma regra, se somente se, novos objetos em $B^\alpha(d_i)$ pertencentes a classe d_i forem cobertos. Caso contrário, o par é descartado para evitar regras redundantes. O processo para quando todas as amostras em $B^\alpha(d_i)$ são cobertas.

O Sistema de Classificação

As regras fuzzy associadas a cada classe d_i , $1 \leq i \leq K \leq \leq$, são avaliadas. A ignorância no processo de classificação é identificada via uma cláusula ELSE que é incluída no sistema de classificação proposto para indicar uma nova classe d_{K+1} : "necessita de complemento". O grau de pertinência μ_i de uma dada amostra a nova classe d_{K+1} , que compõe a cláusula ELSE, é avaliada como $\mu_{k+1} = 1 - \max\{\mu_1, \mu_2, \dots, \mu_k\}$, em que μ_i .

Para uma dada amostra :

- Se regras de classes diferentes tem o mesmo valor máximo para o grau de pertinência μ_i diz-se ocorre igualdade de evidências e a amostra é classificada na classe especial necessita de complemento; neste caso $\mu_{k+1} = 1$.
- Senão, a amostra de teste será associada à classe i , $i = 1, 2, \dots, k, k + 1$ que possuir o maior grau de pertinência.

Exemplo ilustrativo de uso da proposta *CBRU*^α

Para auxiliar no entendimento do método proposto, considere um problema de classificação, cuja base de dados numérica já está transformada em termos linguísticos, veja Tabela 3.1. A base de dados inclui 12 padrões de treinamento, três *atributos condicionais*: {Forma = {redondo, quadrado}, Peso = {leve, médio, pesado} e Frequencia = {baixa, média, alta} } e um *atributo de decisão* (classe) $d_i = d_1, d_2$. Considere para esse exemplo as classes $d_1 = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ e $d_2 = \{x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}$.

Considere todos os possíveis grânulos do conhecimento obtidos como descrito na seção 3.3.2, $RA = \{ B_1 = \{\text{Forma}\}, B_2 = \{\text{Peso}\}, B_3 = \{\text{Frequência}\}, B_4 = \{\text{Forma}, \text{Peso}\}, B_5 = \{\text{Forma}, \text{Frequência}\}, B_6 = \{\text{Peso}, \text{Frequência}\}, B_7 = \{\text{Forma}, \text{Peso}, \text{Frequência}\} \}$. Considere também $\alpha = 0,75$. Devido ao número reduzido de amostras e de atributos da base de dados desse exemplo ilustrativo, todos os possíveis grânulos do conhecimento são usados.

A Tabela 3.3.2 mostra as partições U/B_g geradas e os respectivos $B^{*\alpha=0.75}(d_i)$.

Tabela 3.2: *Rough sets* para as classes X_1 e X_2 considerando $B_1, B_2, B_3, B_4, B_5, B_6$ e B_7 .

$U/B_1 = \{\{x_1, x_2, x_3, x_6, x_7, x_8, x_9, x_{10}, x_{11}\}, \{x_4, x_5, x_{12}\}\}$	
$B_1^{*\alpha=0.75}(d_1) = \{\}$	$B_1^{*\alpha=0.75}(d_2) = \{\}$
$U/B_2 = \{\{x_1, x_4, x_5, x_7\}, \{x_2, x_3, x_6, x_8, x_{10}, x_{11}\}, \{x_9, x_{12}\}\}$	
$B_2^{*\alpha=0.75}(d_1) = \{x_1, x_4, x_5, x_7\}$	$B_2^{*\alpha=0.75}(d_2) = \{x_9, x_{12}\}$
$U/B_3 = \{\{x_1\}, \{x_2, x_3, x_6, x_8, x_{10}, x_{11}, x_{12}\}, \{x_4, x_5, x_7, x_9\}\}$	
$B_3^{*\alpha=0.75}(d_1) = \{1\}$	$B_3^{*\alpha=0.75}(d_2) = \{\}$
$U/B_4 = \{\{x_1, x_7\}, \{x_2, x_3, x_6, x_8, x_{10}, x_{11}\}, \{x_4, x_5\}, \{x_9\}, \{x_{12}\}\}$	
$B_4^{*\alpha=0.75}(d_1) = \{x_4, x_5\}$	$B_4^{*\alpha=0.75}(d_2) = \{x_9, x_{12}\}$
$U/B_5 = \{\{x_1\}, \{x_2, x_3, x_6, x_8, x_{10}, x_{11}\}, \{x_4, x_5\}, \{x_7, x_9\}, \{x_{12}\}\}$	
$B_5^{*\alpha=0.75}(d_1) = \{x_1, x_4, x_5\}$	$B_5^{*\alpha=0.75}(d_2) = \{x_7, x_9, x_{12}\}$
$U/B_6 = \{\{x_1\}, \{x_2, x_3, x_6, x_8, x_{10}, x_{11}\}, \{x_4, x_5, x_7\}, \{x_9\}, \{x_{12}\}\}$	
$B_6^{*\alpha=0.75}(d_1) = \{x_1\}$	$B_6^{*\alpha=0.75}(d_2) = \{x_9, x_{12}\}$
$U/B_7 = \{\{x_1\}, \{x_2, x_3, x_6, x_8, x_{10}, x_{11}\}, \{x_4, x_5\}, \{x_7\}, \{x_9\}, \{x_{12}\}\}$	
$B_7^{*\alpha=0.75}(d_1) = \{x_1, x_4, x_5\}$	$B_7^{*\alpha=0.75}(X_2) = \{x_7, x_9, x_{12}\}$

A partir da Tabela 3.3.2 temos :

$$B^\alpha(d_1) = \{x_1, x_4, x_5, x_7\}$$

$$L_{d_1} = \{(\{x_1, x_4, x_5, x_7\}, B_2), (\{x_1\}, B_3), (\{x_1, x_4, x_5\}, B_5), (\{x_4, x_5\}, B_4)\}$$

$$B^\alpha(d_2) = \{x_7, x_9, x_{12}\}$$

$$L_{d_2} = \{(\{x_9, x_{12}\}, B_2), (\{x_7, x_9\}, B_5), (\{x_{12}\}, B_5)\}$$

Os elementos em $B^\alpha(d_1)$ são obtidos percorrendo todos os 7 grânulos do conhecimento, de B_1 até B_7 . Neste exemplo os elementos x_1, x_4, x_5, x_7 em $B^\alpha(d_1)$ são todos originados de $B_2^{*\alpha=0.75}(d_1)$. Ao percorrer os grânulos do conhecimento restantes nenhum novo elemento foi adicionado a $B^\alpha(d_1)$.

Os elementos em L_{d_1} foram obtidos percorrendo as classes de equivalência obtidas utilizando os 7 grânulos do conhecimento, de B_1 até B_7 .

$B^\alpha(d_2)$ e L_{d_2} foram obtidos de modo similar.

O classificador com o conjunto de regras obtido é :

R_1 : IF *Peso* is *leve* THEN *classe* = d_1

R_3 : IF *Peso* is *pesado* THEN *classe* = d_2

R_4 : IF *Forma* is *redondo* AND *Frequencia* is *baixa* THEN *classe* = d_2

R_5 : *ELSE* *classe* = "necessita de complemento"

Neste exemplo com $\alpha = 0.75$ as amostras x_1, x_4, x_5 são corretamente classificadas na classe d_1 e as amostras x_7, x_9, x_{12} na classe d_2 . As amostras x_2, x_3, x_6 da classe d_1 e as amostras x_8, x_{10}, x_{11} da classe d_2 não são possíveis de serem classificadas pois são indiscerníveis em qualquer grânulo do conhecimento, ou seja, apresentam informações ambíguas.

A amostra x_7 pode ser classificada em ambas as classes d_1, d_2 ou "impossível classificar" dependendo da partição *fuzzy* sendo usada.

Avaliação Experimental

4.1 Materiais e métodos

4.1.1 Bases de dados

Para a avaliação do método proposto foram utilizadas cinco bases de dados fornecidas pela *University of Wisconsin - UCI Machine Learning Repository* [1]. As bases escolhidas foram, Iris, Wine, Wdbc (com 10 características e com 30 características) e Wpbc.

A base de dados Iris é talvez a mais conhecida dentre pesquisadores na área de reconhecimento de padrões. O conjunto de dados contém 3 classes com 50 amostras cada, onde cada classe refere-se a um tipo de planta íris (Iris Setosa, Iris Versicolour e Iris Virginica). Uma das classes é linearmente separável das outras 2. As demais bases de dados não são linearmente separáveis umas das outras.

A base de dados Wine é resultado de uma análise química dos vinhos produzidos na mesma região da Itália, mas provenientes de três diferentes cultivadores. A análise determinou a quantidade de 13 componentes encontrados em cada um dos três tipos de vinhos.

As bases Wdbc e Wpbc são relativas ao câncer de mama. As características das amostras foram extraídas de imagens digitalizadas de células do tecido mamário obtidas por punções com agulha. Elas descrevem as características de núcleos celulares presentes na imagem. Algumas das imagens da base Wdbc podem ser encontradas em [4]. A base Wdbc possui 10 características extraídas e para cada uma delas foram computados o desvio padrão e a média dos 3 maiores valores. Isto resultou em uma base de dados com 30 características. Neste trabalho foram utilizadas as duas versões da base Wdbc, uma com as 10 características e outra com as 30 características. Uma caracterização quantitativa das bases é apresentada na Tabela 4.1.

Os valores numéricos dos *atributos condicionais* das bases de dados foram categorizados como descrito na etapa de transformação dos dados na seção 3.3.1. A Tabela 4.2 mostra os valores utilizados para p (número de funções de pertinência *fuzzy* para categorização dos *atributos condicionais*). Os valores utilizados para α e do suporte para o algoritmo Apriori foram de 0.90 e 7%, respectivamente.

Tabela 4.1: Características das bases de dados utilizadas

x_i	Base de dados	Nº amostras	Nº <i>atributos condicionais</i>	Nº classes
x_1	Iris	150	4	3
x_2	Wine	179	13	3
x_3	Wdbc	569	10	2
x_3	Wdbc	569	30	2
x_4	Wdbc	669	10	2

Tabela 4.2: Número de funções de pertinência *fuzzy* (trapézio), p , utilizadas para categorização dos *atributos condicionais* das bases de dados

Base de dados	p
Iris	4
Wine	6
Wdbc (10 atributos)	8
Wdbc (30 atributos)	8
Wdbc	8

4.1.2 Técnica de validação

Esta é a etapa 5 (Avaliação e interpretação do conhecimento extraído) do processo de KDD. Os experimentos foram executados utilizando validação cruzada *K-fold* (*k-fold cross validation*) [56]. Esta é uma técnica para avaliar os resultados de uma análise estatística de modo generalizado para uma base de dados. Sendo assim é utilizada com frequência para estimar o modo como um modelo preditivo (classificador) irá se comportar na prática. Para reduzir a variabilidade e interferência de hipóteses sugeridas pelos dados [16], várias execuções são realizadas com diferentes partições dos dados em treinamento e teste.

Em uma validação cruzada *K-fold*, o conjunto de dados da base de dados original é particionado em K subconjuntos. Destes K subconjuntos, um único subconjunto é retido como dados teste, e os $K - 1$ subconjuntos restantes são utilizados como dados de treinamento para geração do classificador. O processo de validação cruzada é então repetido K vezes, com cada uma das K subamostras utilizadas exatamente uma vez como dados para teste. Os K resultados obtidos serão então combinados para produzir uma única estimativa, ou seja, serão reportadas as médias dos K valores produzidos para cada uma das métricas. A vantagem deste método sobre uma subamostragem aleatória é que todas as amostras são usadas tanto para treinamento quanto para teste. No caso de uma base de dados desbalanceada, isto é, uma classe com quantidade de amostras diferente da quantidade de amostras de outra classe, em cada *fold* a mesma proporção de amostras é mantida.

4.2 Resultados

4.2.1 Comparação com outros trabalhos

Os resultados foram comparados com outros trabalhos encontrados na literatura, veja Tabela 4.3 (10 X CV significa 10-fold cross validation).

Tabela 4.3: Comparação do classificador proposto com outros métodos.

Base Iris	
Método, autor, avaliação	Precisão
<i>CBRU</i> ^α , 10 X CV	100,00
LS-SVM, Chen e Tsai [37], 10 X CV	98,53
AIRS, Goodman et al. [33]	97,2
CPAR, Yin e Han [63], 10 X CV	94,70
FUZZY CLUSTERING, Abonyi e Szeifert [?], 10 X CV	95,57
NEFCLASS, Nauck e Kruse [10], 10 X CV	95,06
RIAC, Hamilton et al. [27], 10 X CV	94,99
C4.5, Quinlan [54], 10 X CV	94,74
DOEA, Tan et. al. em [7], 50% treino e 50% teste	92,81
Base Wine	
<i>CBRU</i> ^α , 10 X CV	100,00
CFAR, Lu et al. [29], 10 X CV	97,16
CPAR, Yin e Han [63], 10 X CV	95,5
C4.5, Quinlan [53], 10 X CV	92,7
Base Wdbc 1 com 10 características	
<i>CBRU</i> ^α , 10 X CV	99,03
Gaussian process, Seeger [25], 10 X CV	97,03
SVM, Seeger [25], 10 X CV	96,65
Chi, Liu e Setiono [55], 10 X CV	93,22
C4.5, Quinlan [54], 10 X CV	92,48
Linear discriminant, Seeger [25], 10 X CV	92,945
Base Wdbc 2 com 30 características	
<i>CBRU</i> ^α , 10 X CV	98,03
RN, Anagnostopoulos and Maglogiannis [24], 10 X CV	97,90
H-Bspline, Yuan-chin em [8], 10 X CV	97,50
Neighborhood, Qinghua et al. em [22], 10 X CV	96,85
Base Wdbc 3 com 30 características	
<i>CBRU</i> ^α , 10 X CV	93,90
RN, Anagnostopoulos e Maglogiannis [24], 10 X CV	92,80
Nonlinear Classification, Mangasarian e Wild [34], 10 X CV	91,0
Neighborhood, Qinghua et al. [22], 10 X CV	78,82
KBPSVM, 10 X CV, Khemchandani et al. em [51]	68,45

Os resultados mostram um desempenho superior do método proposto com relação aos outros métodos aplicados para as mesmas bases de dados. Note que em todos os casos o método proposto apresenta melhores resultados. A Tabela 4.4 apresenta um resumo do número médio de regras geradas para cada base de dados, número médio de amostras não classificadas e número médio de termos antecedentes na regras. Considere Wdbc 1 e Wdbc 2 como as bases de dados Wdbc com 10 atributos e Wdbc com 30 atributos, respectivamente.

Para obter uma avaliação de precisão justa, foram feitos testes removendo das bases de dados amostras que apresentaram ambiguidade de informações, na etapa de treino, segundo o método proposto. Foram realizados três testes, são eles :

1. Teste 1 : Remoção, para cada uma das 10 validações cruzadas, das amostras não classi-

Tabela 4.4: Resultados em termos de precisão de classificação, número médio de não classificáveis, número médio de termos antecedentes nas regras.

Base Iris							
Base	tx. de tp	tx. de fp	precisão	não classificáveis	num. regras	num. antecedentes	classe
Iris	1,0000	0,0000	1,0000	0,00	1,00	1,00	0
	1,0000	0,0000	1,0000	1,60	3,10	1,51	1
	1,0000	0,0000	1,0000	0,80	3,10	1,41	2
Wine	1,0000	0,0000	1,0000	0,40	9,30	1,77	0
	0,9714	0,0000	1,0000	0,20	8,00	1,61	1
	1,0000	0,0181	0,9714	0,20	7,20	1,55	2
Wdbc 1	0,9968	0,0998	0,9546	3,80	3,30	1,33	0
	0,9001	0,0031	0,9944	5,90	6,90	1,00	1
Wdbc 2	0,9885	0,1842	0,90040	0,00	7,40	1,74	0
	0,8157	0,0114	0,97850	0,00	16,90	1,42	1
Wpbc	0,9797	0,2201	0,9047	0,30	23,50	2,22	0
	0,7798	0,0202	0,9571	2,60	44,70	1,67	1

ficáveis no treino e teste com as amostras de treino originais.

2. Teste 2 : Remoção, para cada uma das 10 validações cruzadas, das amostras não classificáveis no treino e teste com as amostras de teste originais.
3. Teste 3 : União das amostras não classificáveis no treino em todas as 10 validações cruzadas . Remoção destas amostras resultantes da base de dados original de modo a obter uma segunda base de dados sem as amostras não classificáveis no treino. Realização de validação 10-*fold* normalmente para esta segunda base de dados obtida.

No Teste 1 foi avaliado se o número de amostras não classificáveis é alterado ao treinar com a base de treino sem as amostras não classificáveis no treino inclusas e testar com a base treino com as amostras não classificáveis no treino inclusas. Os resultados mostraram que o número de amostras não classificáveis no treino foi mantido em todas as bases de dados.

Os resultados para as bases de dados Iris, Wine, Wdbc e Wpbc são mostrados nas Tabelas 4.5, 4.6, 4.7 e 4.8, respectivamente. Note que o classificador proposto neste trabalho mantém a melhor taxa de precisão de classificação para o Teste 2 e para o Teste 3.

Estes resultados mostram que o classificador realmente não sofre influência das amostras não classificáveis no treino.

4.2.2 Impacto de redução de dimensionalidade no método proposto e em outros métodos

Para verificar se a redução prévia de dimensionalidade impacta no resultado do classificador proposto neste trabalho foram realizados 3 testes. No primeiro teste foi aplicado à base de dados original uma transformação do espaço característica utilizando Análise de Componentes Principais (PCA) e utilizando todas as componentes principais. No segundo teste foi utilizando apenas a metade das componentes principais. Em um terceiro teste foi aplicado um algoritmo

Tabela 4.5: Resultados em termos de precisão de classificação relativa a base de dados Iris para o Teste 2 e Teste 3.

Base Iris		
Método	Teste 2	Teste 3
<i>CBRU</i> ^α	1,0000	1,0000
Bayes Net	0,9483	0,9643
Decision Table	0,9505	0,9563
J48	0,9510	0,9563
SVM	0,95,10	0,9256
Rede Neural MultiLayer Perceptron	0,9510	0,9696

Tabela 4.6: Resultados em termos de precisão de classificação relativa a base de dados Wine para o Teste 2 e o Teste 3.

Base Wine		
Método	Teste 2	Teste 3
<i>CBRU</i> ^α	0,9904	0,9910
Bayes Net	0,9769	0,9876
Decision Table	0,9462	0,9770
J48	0,9498	0,9730
SVM	0,9822	0,9650
Rede Neural MultiLayer Perceptron	0,9769	0,9900

de redução de atributos disponível na plataforma Rosetta [2]. Esse algoritmo verifica de modo exaustivo se é possível subconjuntos mínimos de *atributos condicionais* (redutos de atributos definidos na seção 2.3.2) mantendo a mesma classificação para as amostras do universo do discurso [67]. Duas bases de dados apresentaram redutos de atributos com uma quantidade de atributos menor que o número de atributos original. A base Wdbc com 30 atributos teve sua dimensionalidade reduzida para 16 atributos e a base de dados Wine com 10 atributos reduziu a dimensionalidade para 7 atributos .

Para o método proposto a redução de dimensionalidade não melhorou os resultados em termos de precisão, número de regras, número de termos antecedentes e número de amostras não classificáveis para nenhuma das bases de dados, exceto para a base Wpbc, na qual a

Tabela 4.7: Resultados em termos de precisão de classificação relativa a base de dados Wdbc com 10 características para o Teste 2 e o Teste 3.

Base Wdbc com 10 características		
Método	Teste 2	Teste 3
<i>CBRU</i> ^α	0,9745	0,9776
Bayes Net	0,9684	0,9665
Decision Table	0,9378	0,9600
J48	0,9368	0,9670
SVM	0,9646	0,9660
Rede Neural MultiLayer Perceptron	0,9530	0,9860

Tabela 4.8: Resultados em termos de precisão de classificação relativa a base de dados Wpbc para o Teste 2 e o Teste 3.

Base Wpbc		
Método	Teste 2	Teste 3
<i>CBRU</i> ^α	0,9309	0,9312
Bayes Net	0,9218	0,9360
Decision Table	0,9128	0,9465
J48	0,9284	0,9435
SVM	0,9356	0,9410
Rede Neural MultiLayer Perceptron	0,9331	0,9580

transformação/redução da base de dados utilizando PCA melhorou os resultados com relação a estes aspectos. Provavelmente as amostras desta base de dados, originalmente, tem seus valores bastante próximos para as diversas classes, o que reduz a capacidade de discernibilidade entre as amostras. A aplicação do PCA pode ter "alongado" as distâncias entre os valores das características das amostras. A utilização de metade das componentes principais (igual a metade do número de características na base de dados original) não traz melhorias significativas se comparada com a utilização de todas as componentes (igual ao número de características na base de dados original) nesta base de dados. Com base nos resultados apresentados a redução de dimensionalidade não traz impacto na precisão de classificação do método proposto.

4.2.3 Discussão dos resultados

Os resultados experimentais mostram um desempenho superior do método proposto com relação aos outros métodos aplicados para as mesmas bases de dados. Para investigar se o método sofreria melhorias com uma redução prévia de dimensionalidade foram realizados testes utilizando a transformação de dimensionalidade utilizando PCA (*Principal Component Analysis*) e os redutos definidos na TRS. Os resultados mostram que o método proposto não é impactado por esta redução prévia de dimensionalidade, no entanto é prematuro afirmar que o método já faz implicitamente uma seleção de atributos, embora esta conclusão seja intuitiva.

Outra investigação realizada foi a verificação se as amostras não classificáveis impactam ou não na geração das regras e no resultado do classificador. Para tanto, as amostras consideradas não classificáveis foram retiradas das bases de dados. Os testes mostraram que, para todas as bases de dados, a precisão de classificação no teste foi exatamente a mesma que a obtida ao treinar com as amostras não classificáveis inclusas. Estes resultados mostram que estas amostras não classificáveis são realmente reconhecidas e não impactam no método de geração de regras.

A redução do espaço de característica via transformação da dimensionalidade utilizando a metade das componentes principais obtidas ao aplicar a técnica de PCA e a redução de dimensionalidade via a redução de atributos utilizando a plataforma *Rosetta* não impactaram o método proposto em termos de precisão de classificação, número de regras geradas, número de termos antecedentes e número de amostras não classificáveis. Esta análise nos leva a intuir que o método proposto neste trabalho faz implicitamente uma redução de dimensionalidade via

seleção de características.

Com o objetivo de verificar se o método proposto realmente gerou regras concisas, cada conjunto de antecedentes das regras geradas foi submetido a um algoritmo de redução de atributos disponível na plataforma Rosetta [2]. O algoritmo verifica de modo exaustivo se é possível reduzir o número de antecedentes de uma regra sem prejuízo da dependência parcial de atributos [67]. Não houve redução de antecedentes para nenhuma das regras previamente obtidas.

Conclusão

5.1 Principais contribuições

Este trabalho propôs um método para geração automática de regras if-then *fuzzy* baseado nas teorias de fuzzy sets e rough sets. Este trabalho contribui com a proposta de uma extensão para operações de aproximação definidas na TRS, denominada aproximação *upper^{alpha}* de X , e denotada por $B^{*\alpha}(X)$, a fim de adequar os Rough Sets aos propósitos deste trabalho.

A contruição mais importante deste trabalho é o método de geração de regras proposto, o qual garante a produção automática das regras *fuzzy* do tipo if-then com número reduzido de antecedentes e com alta taxa de cobertura quanto aos objetos do universo do discurso.

As regras *fuzzy* geradas formam o núcleo do classificador proposto, o qual é capaz de discernir quanto a possibilidade ou impossibilidade de classificação. Aquelas amostras que não apresentam padrão para serem classificadas em nenhuma das classes do domínio da aplicação são classificadas como "necessita de complemento".

O classificador foi avaliado utilizando bases de dados públicas do UCI Machine Learning Repository [1]. Os resultados obtidos, em termos de precisão, foram superiores aos resultados apresentados por outros métodos utilizando as mesmas base de dados.

5.2 Publicações

Como fruto deste trabalho, até o presente momento, foram publicados os dois artigos [31] e [18].

5.3 Trabalhos futuros

O conjunto de regras obtido pelo método proposto está fortemente ligado a partição *fuzzy* gerada. Os trabalhos futuros incluem o desenvolvimento de um método para obtenção automática da partição *fuzzy* levando em conta a distribuição dos dados e a realização de testes em bases de dados mais volumosas.

Outra questão a ser abordada é a adaptação do método proposto para determinar o grau de ocorrência de ruídos e *outliers* em bases de dados.

Referências Bibliográficas

- [1] Asuncion A and Newman D J. <http://www.ics.uci.edu/~mlearn/MLRepository.html>. UCI Machine Learning Repository - School of Information and Computer Sciences, 2007.
- [2] Ohrn A, Komorowski J, Skowron A, and Synak P. The design and implementation of a knowledge discovery toolkit based on rough sets - the rosetta system. *Rough Sets in Knowledge Discovery*, 1998.
- [3] Zadeh L A. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–23, 1978.
- [4] Unknown Author. <http://pages.cs.wisc.edu/street/images/>. *The University of Winsconsin*. Acessado em 20 de junho, 2009, 2009.
- [5] Liu B, Hsu W, and Ma Y. Integrating classification and association rule mining. *KDD'98*, pages 80–86, 1998.
- [6] Tan K C, Tay A, Lee T H, and Heng C M. Mining multiple comprehensible classification rules using genetic programming. *CEC '02*, 2:1302–1307, 2009.
- [7] Tan K C, Yu Q, and Ang J H. A dual-objective evolutionary algorithm for rules extraction in data mining. *Computational Optimization and Applications*, 34(2):273–294, 2006.
- [8] Yuan chin I C. Boosting svm classifiers with logistic regression. *"Journal"Desconhecido*, 2003.
- [9] MAIA G D. *Embriologia Humana*. Atheneu, 5nd edition, 2007.
- [10] Nauck D and Kruse R. Learning in neuro-fuzzy systems with symbolic attributes and-missing values. *Intelligent Syst. Res. Group*, 1:142–147, 1999.
- [11] Pablo Alberto Dalbem de Castro and Heloisa de Arruda Camargo. A study of the reasoning methods impact on genetic learning and optimization of fuzzy rules. In *SBIA*, pages 414–423, 2004.

- [12] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2000.
- [13] Cintra M E and Camargo H A. Fuzzy rules generation using genetic algorithms with self-adaptive selection. *IEEE International Conference on Fuzzy Systems*, 13:261–266, 2007.
- [14] Cintra M E and de Camargo H A. Fuzzy rules generation with pre-selection of candidate rules. *submitted for publication*, 2007.
- [15] Rumelhart D E, McClelland J L, and PDP Research Group. Parallel distributed processing. *MIT*, 1986.
- [16] Mosteller F. A k-sample slippage test for an extreme population. *Annals of Mathematical Statistics*, 19:58–65, 2009.
- [17] Klir GJ and Yuan B. *Fuzzy Sets and Fuzzy Logic - Theory and Applications*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1995.
- [18] Denise Guliato and Jean Carlo Sousa Santos. Granular computing and rough sets to generate fuzzy rules. In *ICIAR '09: Proceedings of the 6th International Conference on Image Analysis and Recognition*, pages 317–326, Berlin, Heidelberg, 2009. Springer-Verlag.
- [19] Ishibuchi H, Nakashima T, and Murata T. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man, And Cybernetic - Part B: Cybernetics*, 29(5):601–618, October 1999.
- [20] Ishibuchi H and Nojima Y. Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning*, 44:4–31, 2007.
- [21] Ishibuchi H, Nojima Y, and Kuwajima I. Genetic rule selection as a postprocessing procedure in fuzzy datamining. In *International Symposium on Evolving Fuzzy Systems*, pages 286–291, September 2006.
- [22] Qinghua H, Daren Y, and Zongxia X. Neighborhood classifiers. *Expert Systems with Applications*, 34:866–876, 2008.
- [23] Tzung-Pei Hong, Chun-E Lin, Jiann-Horng Lin, and Shyue-Liang Wang. Learning cross-level certain and possible rules by rough sets. *Expert Syst. Appl.*, 34(3):1698–1706, 2008.
- [24] Anagnostopoulos I and Maglogiannis I. Neural network-based diagnostic and prognostic estimations in breast cancer microscopic instances. *Med Bio Eng Comput*, 44:773–784, 2006.
- [25] Williams C K I and Seeger M. The effect of the input density distribution on kernel-based classifiers. *ICML*, pages 1159–1166, 2000.

- [26] Gordon J and Shortliffe E H. The dempster-shafer theory of evidence. *Rule-based expert systems*, pages 272–292, 1984.
- [27] Hamilton H J, Shan N, and Cercone N. Riacc: a rule induction algorithm based on approximate classification. *Tech. Rep. CS, Regina University*, pages 96–106, 1996.
- [28] Jesus M J, Gonzalez P, and Herrera F. Multiobjective genetic algorithm for extracting subgroup discovery fuzzy rules. *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*, pages 50–57, 2007.
- [29] Lu J, Xu B, , and Yang H. A classification method of fuzzy association rules. *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 4:248–251, 2003.
- [30] Bezdek JC. *Pattern Recognition with Fuzzy Objective Function Algorithm*. Plenum Press, New York, 1981.
- [31] SANTOS JCS and GULIATO D. Proposta de um método para geração automática de regras fuzzy baseada na teoria dos rough sets. *Latino America de Informática - Clei'09*, 2009.
- [32] Jain A K, Duin R P W, and Mao J. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 200.
- [33] Goodman D L, Boggess L C, and Watkins A B. Artificial immune system classification of multiple-class problems. *Artificial Neural Networks In Engineering, ANNIE*, 1, 2002.
- [34] Mangasarian O L and Wild E W. Nonlinear knowledge-based classification. *Neural Networks, IEEE Transactions on*, 19:1826–1832, 2008.
- [35] Zadeh L. Fuzzy sets and systems. *Fox J, editor. System Theory*, pages 29–39, 1965.
- [36] CHAUI M. Convite à filosofia. *Editores Ática*, 1999.
- [37] Chen S M and Tsai F M. A new method to construct membership functions and generate fuzzy rules from training instances. *References International Journal of Information and Management Sciences*, 16:437–442, 2006.
- [38] Fayyad U M. Data mining and knowledge discovery: making sense out of data. *IEEE Expert*, pages 20–25, 1996.
- [39] Sá J P M. Pattern recognition: Concepts, methods and applications. *Springer*, 2001.
- [40] Sarkar M. Fuzzy-rough nearest neighbor algorithms in classification. *Fuzzy Sets and Systems*, 158:2134–2152, 2007.
- [41] Srinivas M and Patnaik L M. Genetic algorithms: a survey. *Computer*, 27:1–37, 1992.

- [42] Stigler S M. Thomas bayes bayesian inference. *Journal of the Royal Statistical Society*, 145:250–258, 1982.
- [43] Sugeno M. An introductory survey of fuzzy control. *information Science*, 36(1/2):59–83, 1985.
- [44] Zheng D X M, Thomas S N, and Kumaraswamy M M. Applying pareto ranking and niche formation to genetic algorithm based multiobjective time cost optimization. *Constr. Engrg. and Mgmt.*, 131(1):81–91, 2005.
- [45] Hong T P, Wang T T, and Wang S L. Knowledge acquisition from quantitative data using the rough-set theory. *Intelligent Data Analysis*, 4:289–304, 2000.
- [46] Hong T P, Wang T T, and Wang S L. Learning a coverage set of maximally general fuzzy rules by rough sets. *Exepert Systems with Applications*, 19:97–103, 2000.
- [47] Tan PN, Steinbach M, and Kumar V. Introduction to data mining. *Addison- Wesley*, 2006.
- [48] Shen Q and Chouchoulas A. A rough-fuzzy approach for generating classification rules. *Pattern Recognition*, 35:2425 – 2438, 2002.
- [49] Agrawal R and Srikant R. Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, 1994.
- [50] Jensen R and Shen Q. Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions of Knowledge and Data Engineering*, pages 1457–1471, 2004.
- [51] Khemchandani R, Jayadeva J, and Chandra S. Knowledge based proximal support vector machines. *European Journal of Operational Research*, 2009.
- [52] Pal N R and Bezdek J C. On cluster validity for the fuzzy c-means model. *IEEE Trans. Fuzzy Systems*, pages 330–379, 1995.
- [53] Quinlan J R. C4.5: programs for machine learning. *Morgan Kaufmann Publishers Inc.*, 1993.
- [54] Quinlan J R. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Resea*, 4:77–90, 1996.
- [55] Setiono R and Liu H. Neurolinear: From neural networks to oblique decision rules. *Neurocomputing*, 17:1–24, 1997.
- [56] Duda RO, Hart PE, and Stork DG. *Pattern Classification*. Wiley, New York, NY, 2nd edition, 2001.
- [57] Theodoridis S and Koutroumbas K. *pattern recognition*. elsevier academic, 2003.

- [58] Hong TP and Lee YC. Mining coverage-based fuzzy rules by evolutionary computation. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 218–224, Washington, DC, USA, 2001. IEEE Computer Society.
- [59] Cohen W. Fast efective rule induction. *ICML'95*, pages 115–123, 1995.
- [60] Li W, Han J, and Pei J. Cmar: Accurate and efficient classification based on multiple class-association rules. *ICDM'01*, pages 369–376, 2001.
- [61] Liao T W, Celmins A K, and Hammell R J. A fuzzy c-means variant for the generation of fuzzy term sets. *Fuzzy Sets and Systems*, 135:241–257, 1997.
- [62] Wilson S W. Classifier systems and the animat problem. *Machine Learning*, 2:199–228, 1987.
- [63] Yin X and Han J. Cpar: Classification based on predictive association rules. *Third SIAM Int'l Conf. Data Mining*, 2003.
- [64] Cao Y, Liu S, Zhang L, Qin J, Wang J, and Tang K. Prediction of protein structural class with rough sets. *BMC Bioinformatics*, 7:20, 2006.
- [65] Ching J Y, Wong A K C, and Chan K C C. Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on PAMI* 17, pages 641–651, 1995.
- [66] Pawlak Z. Rough sets. *International Journal of Computer and Information Sciences*, 11:341–356, 1982.
- [67] Pawlak Z and Skowron. Rudiments of rough sets. *Information Science*, 177:3–27, 2007.