Detecção de Fraudes em Unidades Consumidoras de Energia Elétrica Usando Rough Sets

José Edison Cabral Junior

Detecção de Fraudes em Unidades Consumidoras de Energia Elétrica Usando Rough Sets

José Edison Cabral Junior

Orientador: ***Prof. Dr. João Onofre Pereira Pinto***

Dissertação apresentada ao Departamento de Engenharia Elétrica da Universidade Federal de Mato Grosso do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

UFMS - Campo Grande
Maio/2005

Detecção de Fraudes em Unidades Consumidoras de Energia Elétrica Usando Rough Sets

José Edison Cabral Junior

Dissertação de Mestrado submetida à banca examinadora designada pelo Colegiado do Programa de Mestrado em Engenharia Elétrica da Universidade Federal de Mato Grosso do Sul, como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 25 de maio de 2005 por:

João Onofre Pereira Pinto - Dr.
Prof. DEL/UFMS - Orientador

Luciana Cambraia Leite - Dra.
Profa. DEL/UFMS

Roberto Navarro de Mesquita - Dr.
Pesquisador IPEN

Kathya Silvia Collazos Linares - Dra.
Pesquisadora DEL/UFMS

Aos meus familiares, amigos, professores e a minha namorada.

Agradecimentos

Agradeço primeiramente ao Pai Celestial, por sempre iluminar minha mente nas encruzilhadas da vida, fazendo com que minhas decisões sejam na maioria das vezes acertadas.

Aos meus pais, Cabral/Enói, minhas irmãs, Luciana/Juliana, e meu sobrinho Marlon, que me apoiaram muito em meus estudos. A minha namorada, Débora, que me acompanha há 2 anos de muitas felicidades.

Ao meu orientador, Prof. João Onofre, que além de contribuir intelectualmente, tornou-se um grande amigo. A sua esposa, Alexandra, pelas revisões deste trabalho.

A todos meus familiares de Juiz de Fora e Coronel Fabriciano, em Minas Gerais, que a despeito da distância, sempre demonstraram muito carinho e apoio.

Também não poderia deixar de agradecer a vários amigos, tanto da vida quanto da universidade: João/Paulo Pegolo, Rodrigo Araújo, Rodrigo Jorge, Guilherme/Henrique/Rafael Brito, Matheus Borges, Aritani Costa, Pedro Bastos, Tatiana Marin, Danilo Viana, Manoel Honda, Rogério Colman, Otávio Lemos, Bruno Gíglio, Jonusi Goiânia, Lorena Robaldo, Sílvia Rodrigues, Leopoldo Lopes, Renato Fischer, Thiago Bueno, Augusto Medina, Profa. Luciana Cambraia, Prof. Milton Romero, Prof. Evandro Mazina, Profa. Bernadete Zanusso, Profa. Kathya Linares, Prof. Jorge Ortiz, Gilberto Tatibana, Edgar Gontijo, Márcio Portela, José Reis, Inez Lino, Cássima Ortegosa, Aldo Alvarenga, Faete Jacques, Luigi Galotto, Ruben Godoy, Cristian Mara, João Okumoto, Evânio Zorzate, Rafael Tramontini, Anderson Teruya, Dionísio Sant'Ana, André Muniz, João Vitor, Wagner Peron, Guilherme Berthier, Leandro Tortosa, Gustavo Henrique, Fábio Costa, Bruno Padovan, Tiago Jorge, Carlos Cézar, Pedro Honda, Maxwell Lima, Alexandre Maeda, Gabriela Garcia, Weber Diniz, Anderson/Amílton Novaes, Cristiano Argemon, Luiz Marchetti e Robert Keele. Peço desculpas se esqueci de alguém, afinal felizmente a minha lista de amigos é grande.

Finalmente, agradeço à CAPES pelo apoio financeiro.

E eu que olhei vi, em disparada agora,
um lábaro que parecia sujeito
a rodear sem pouso e sem demora;

imensa turba o seguia, que o conceito
deu-me, numa visão medonha e abstrusa,
de quantos tinha a morte já desfeito.

(...)

Certo então fui, no entendimento meu,
que o abjeto grupo aquele era da gente
que a Deus despraz e ao inimigo seu.

Esses, de quem foi sempre a vida ausente,
estavam nus, às picadas expostos
de uma nuvem de vespas renitente,

que lhes fazia riscar de sangue os rostos,
que, às lágrimas mesclado, a seus pés
colhiam molestos vermos ali postos.

(Dante Alighieri - A Divina Comédia - Inferno - Canto III)

Resumo

As fraudes representam as maiores perdas comerciais das empresas de distribuição de energia elétrica. Devido ao elevado número de consumidores, as inspeções geralmente são feitas sem uma pré-análise de comportamento dos inspecionados, resultando em baixas taxas de acerto. Como as empresas de distribuição possuem muitas informações sobre seus consumidores armazenadas em bancos de dados, é possível identificar o perfil dos clientes fraudadores e utilizar este conhecimento na orientação das futuras inspeções.

Este trabalho propõe uma metodologia baseada em Rough Sets e KDD para detecção de fraudes em consumidores de energia elétrica. Esta metodologia realiza uma avaliação detalhada da região de fronteira entre clientes normais e fraudadores, identificando padrões de comportamento fraudulentos nos dados históricos das empresas de energia elétrica. A partir destes padrões, derivam-se regras de classificação que, em futuros processos de inspeção, indicarão quais clientes apresentam perfis fraudulentos. Com inspeções guiadas por comportamentos suspeitos, aumenta-se a taxa de acerto e a quantidade de fraudes detectadas, diminuindo as perdas com fraudes nas empresas de distribuição de energia elétrica.

Abstract

Frauds represent a high percentage of the total commercial losses for electrical energy companies. In general, due to the high number of consumers, in-site inspections are made without any criteria, which cause a low rightness rate. On the other hand, electrical energy companies have information about their consumers stored in their databases. This information could be used to identify behavior patterns that are common among consumers that commit frauds, and this could guide the selection of the consumer that should undergo inspection.

This work proposes a KDD and Rough Sets based methodology for consumer fraud detection for electrical energy companies. This methodology helps to find out consumer fraud behavior profiles at the company databases. From these patterns, a set of classification rules are created to fetch consumers that should be inspected. Using such strategy, the companies expect to increase the rightness rate and therefore decrease profit losses due to consumer fraud.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO

1

Introdução

1.1 Contextualização

Um dos grandes problemas enfrentado pelas empresas de distribuição de energia elétrica são as perdas comerciais, ocasionadas principalmente pelas fraudes em unidades consumidoras. Para diminuir as perdas, estas empresas geralmente utilizam-se das inspeções *in loco* para detectar tais fraudes.

As inspeções são feitas por técnicos que visitam as unidades consumidoras e avaliam os instrumentos e ligações elétricas. Geralmente, a empresa possui especialistas que indicam quais unidades consumidoras devem ser alvo de inspeção. Esta decisão baseia-se em fatores como: localidade com média de consumo baixa ou alta incidência de fraude, denúncias, entre outros. Pela grande quantidade de unidades consumidoras, é praticamente impossível o especialista avaliar o comportamento de cada unidade e indicar aquelas com suspeitas de fraude. Também é inviável inspecionar todas as unidades consumidoras, visto que o número de fraudadores é pequeno em relação ao total de clientes. A taxa de acerto das empresas de distribuição de energia elétrica varia de 5 à 10%.

Porém, sabe-se que as empresas de distribuição de energia elétrica armazenam informações de seus clientes em banco de dados. Estas informações podem ser utilizadas para a identificação de padrões ou perfis de comportamento. Encontrando um perfil que indique um comportamento fraudulento, o especialista pode recomendar que os clientes com este perfil sejam inspecionados. O processo de descoberta destes padrões de comportamento à partir de banco de dados deve ser realizado de forma automática, por alguma ferramenta computacional que analise os dados e extraia o conhecimento.

As técnicas de Inteligência Artificial (IA), por sua vez, procuram incrementar habilidades do ser humano aos sistemas computacionais. Estas habilidades podem ser: a tomada de decisão em situações desconhecidas, o reconhecimento de sinais, a capacidade de aprender, entre outros. Uma das principais vertentes da IA é a Descoberta de Conhecimento em Banco de Dados, ou *Knowledge Discovery in Database* (KDD).

A *Teoria de Rough Sets*[1], considerada também uma técnica de IA, possui como base o não-discernimento entre dados, utilizando-se dos conceitos de conjuntos finitos e suas cardinalidades, além de relações e classes de equivalência. Visto que perfis de comportamento fraudulento são sutis e assemelham-se muito com perfis normais, a teoria de Rough Sets (como ferramenta de KDD) possui aplicabilidade direta no problema em questão.

Este trabalho apresenta uma metodologia baseada em Rough Sets para detecção de fraudes em consumidores de energia elétrica. A aplicação desta metodologia identifica padrões de comportamento fraudulentos em bancos de dados de empresas de energia elétrica. A partir destes padrões, derivam-se regras de classificação que, em futuros processos de inspeção, indicarão quais clientes apresentam perfis fraudulentos. Com inspeções guiadas por comportamentos suspeitos, aumenta-se a taxa de acerto e a quantidade de fraudes detectadas, diminuindo as perdas com fraudes nas empresas de distribuição de energia elétrica.

1.2 Revisão Bibliográfica

A fraude é um crime que ocorre nas mais diversas áreas e atividades ao redor do globo: cartões de crédito, seguros (automóveis, imobiliários, etc), planos de saúde, imposto de renda, telefonia (fixa e móvel), bancos, postos de combustíveis, consumos de água e energia, etc.

Uma pesquisa, envolvendo aproximadamente 1.000 empresas brasileiras do ramo industrial, revelou que somente 50% das perdas por fraudes no ano de 2004 foram recuperadas (KPMG, 2004). Segundo a mesma pesquisa, 71% das empresas indicaram a precariedade do sistema de controle interno como a principal circunstância facilitadora de fraudes. Mesmo que as empresas tenham respondido a pesquisa de modo a amenizar suas verdadeiras perdas por fraude, os números apresentados indicam a gravidade do problema e a necessidade de mecanismos de detecção mais eficientes.

Existem disponíveis no mercado alguns programas comerciais para detecção de fraudes. O Clementine[2], comercializado por SPSS Inc., disponibiliza ferramentas de classificação, agrupamento e predição, podendo ser utilizado na detecção de vários tipos de fraudes. Implementa árvores de decisão, redes neurais artificiais, dentre outras técnicas. Já o programa Falcon Fraud

[1] A tradução para Rough Sets seria Conjuntos "Aproximados", ou "Incertos", porém os termos Rough Sets ou Teoria de Rough Sets são mais aceitos e utilizados na literatura

[2] http://www.spss.com/clementine/

Manager, comercializado por Fair Isaac[3], utiliza modelos de redes neurais artificiais para detecção de fraudes em cartões de débito e crédito.

A detecção de fraudes em cartões de crédito concentra a maioria dos trabalhos publicados. Em (Kou et al., 2004) encontra-se uma revisão dos principais métodos contra fraudes em cartões de crédito, invasão de computadores e fraudes em telecomunicações. O uso ilícito de cartões de crédito é dividido em fraudes *offline* e *online*. A primeira considera a utilização de um cartão de crédito falso ou roubado e ainda não bloqueado para uso. A segunda consiste do uso de cartões de crédito ilícitos na realização de compras via internet ou telefone, onde não se exige assinatura manual.

As técnicas e metodologias para detecção de fraudes em cartões de crédito são baseadas no histórico completo de transações dos portadores ou apenas nas informações recentes e inerentes a uma nova transação. O modo de aprendizado destas técnicas pode ser supervisionado ou não-supervisionado. No aprendizado supervisionado, comportamentos fraudulentos em históricos de transações são "minerados" e comparados a cada nova transação, na busca por operações ilícitas pré-concebidas. A desvantagem do aprendizado supervisionado é a dificuldade em detectar comportamentos fraudulentos não encontrados previamente nos históricos de transações (Bolton e Hand, 2001). Já no aprendizado não-supervisionado, encontra-se o comportamento normal de cada portador de cartão de crédito à partir de seu histórico de transações. Variações na freqüência ou no valor das transações, por exemplo, podem apontar desvios do comportamento normal, indicando possíveis fraudes (Hung e Cheung, 1999). A principal adversidade deste método é controlar o número de falsos alarmes, ou seja, diferenciar transações legais incomuns (exceções) de transações fraudulentas.

O CARDWATCH é um programa de mineração de dados voltado para a detecção de fraudes em cartões de crédito, baseando-se em uma rede neural artificial alimentada adiante (*feed-forward*) (Aleskerov et al., 1997). A partir de dados gerados por simulações (dados artificiais), alcançou-se uma taxa de acerto de 85% na detecção de fraudes.

Em (Dorronsoro et al., 1997) é apresentado um sistema para detecção de fraudes em cartões de crédito que atua entre o local de compra e o acesso às operadoras (VISA[4], MasterCard[5], American Express[6], etc). Este sistema utiliza uma rede neural artificial do tipo perceptron de múltiplas camadas funcionando como um classificador de transações. A rede neural possui: uma camada de entrada, que recebe as variáveis do sistema; várias camadas ocultas que transformam as variáveis de forma não-linear; uma camada de saída contendo $C - 1$ neurônios (onde C é o número de classes de decisão), funcionando como um analisador de discriminantes não-linear. A função de ativação dos neurônios é do tipo sigmoidal, com exceção das unidades que ocupam a última camada oculta, que utilizam função linear. O objetivo geral do sistema

[3]http://www.fairisaac.com/fairisaac
[4]http://www.visa.com/
[5]http://www.mastercard.com/
[6]http://www.americanexpress.com/

é encontrar o modelo do comportamento normal de cada cliente à partir de informações correntes e imediatamente anteriores à operação (transação). Desvios do comportamento normal geram alertas para as operações consecutivas, as quais podem ser bloqueadas por suspeita de fraude. O sistema, portanto, não considera o histórico completo de transações de cada cliente, privilegiando o tempo de resposta da avaliação, que fica em torno de 60 ms. A partir de um conjunto de teste, o sistema alcançou uma taxa de acerto de fraude em torno de 50%. No ano de publicação, este sistema estava em operação na Espanha e avaliava mais de 12 milhões de transações por ano.

Em (Kwon e Feroz, 1996) também é utilizada uma rede neural artificial do tipo perceptron de múltiplas camadas, porém com o propósito de identificar fraudes em relatórios financeiros de empresas do setor industrial. A partir de um histórico, foram selecionadas 70 empresas com relatórios suspeitos, dos quais 35 realmente continham fraude. A partir de informações destes relatórios, foram criadas 19 variáveis para cada empresa, compondo os dados a serem submetidos à rede neural. O conjunto de treinamento contou com 55 (79%) empresas e o conjunto de teste com 15 (21%), ambos selecionados aleatoriamente. O teste apresentou uma taxa de acerto de 88%, contra 47% de outras ferramentas de mesmo propósito.

No trabalho (Passini, 2002) foi utilizado o programa DB2 Intelligent Miner, comercializado pela IBM[7], na mineração de dados para a detecção de fraudes em ligações de água. O resultado esperado previamente não foi alcançado: a diminuição de 51 para 41% na porcentagem de inspeções *in loco* com resultado negativo (não detecção de fraude).

Em (Henriques et al., 2001) é proposta uma metodologia para estimação e localização de perdas comerciais utilizando redes neurais e conjuntos nebulosos (*fuzzy sets*), visando a identificação de fraudes. Para tanto, é dito que a metodologia utiliza informações de consumo de energia e de medições realizadas nas subestações. Porém, o trabalho apresenta somente um resumo das técnicas citadas acima, não apresentando detalhes suficientes para qualquer avaliação da metodologia. Nenhum resultado previsto ou estimado é apresentado no trabalho.

Foram encontradas somente duas publicações abordando a detecção de fraudes em consumidores de energia elétrica. No trabalho (Cabral et al., 2004) foram utilizados alguns conceitos de Rough Sets para a identificação de padrões de comportamento fraudulentos em dados históricos. Um conjunto de clientes e seus respectivos atributos foram organizados em um Sistema de Informação, onde foram aplicados os conceitos de aproximação inferior, reduto e do algoritmo da decisão mínima, ou *minimal decision algorithm* (MDA). A partir do Sistema de Informação reduzido, derivou-se um conjunto de regras as quais representaram perfis de comportamento de clientes. Tomando-se os perfis de comportamento fraudulento, consolidou-se um sistema de regras de classificação, o qual alcançou uma taxa de acertos de fraude de 20%.

Em (Reis et al., 2004) é apresentado um sistema de pré-seleção de consumidores de energia elétrica para inspeção, com o objetivo de detectar fraudes e erros de medição. A partir do banco

[7] http://www-306.ibm.com/software/data/iminer/

de dados de uma empresa de distribuição de energia elétrica, foram selecionados 5 atributos (dentre os 52 disponíveis) e 40.000 registros (de um total de 600.000). O sistema é baseado em uma árvore de decisão CART (Breiman et al., 1993), a qual foi treinada com 20.000 registros selecionados aleatoriamente. O teste do sistema com os 20.000 registros remanescentes resultou em uma taxa de acerto de 40% para fraudadores, 35% a mais que a taxa alcançada pela empresa em questão.

1.3 Objetivos

Este trabalho tem como principais objetivos:

1. Apresentar a teoria de Rough Sets como técnica de Inteligência Artificial, através de abordagens prática e teórica;
2. Aplicar a teoria de Rough Sets em um problema real de descoberta de conhecimento em banco de dados.
3. Apresentar uma metodologia para a detecção de fraudes em unidades consumidoras de energia elétrica, a qual seja aplicável à bancos de dados de diferentes empresas de distribuição de energia elétrica.

1.4 Organização do Trabalho

No Capítulo 2 é apresentada uma visão geral do que é Inteligência Artificial e Aprendizado de Máquina, enfocando o processo de KDD e cada uma de suas etapas. Já no Capítulo 3 são apresentados os principais conceitos de Rough Sets, objetivando dar ao leitor uma idéia geral desta emergente técnica. No Capítulo 4 é discutida a fundamentação teórica de Rough Sets, consolidando os conceitos do Capítulo 3. A partir dos dados da empresa de distribuição de energia elétrica, o pré-tratamento e a consolidação dos dados são feitos no Capítulo 5. Finalmente, no Capítulo 6, é apresentada uma metodologia para a detecção de fraudes em consumidores de energia elétrica, a qual é baseada nos conceitos de Rough Sets abordados neste trabalho.

CAPÍTULO

2

Inteligência Artificial, Aprendizado de Máquina e KDD

2.1 Introdução

A informação ocupa hoje o mais elevado patamar da corrida tecnológica. Chama-se de *tecnologia da informação* (TI) a utilização de dispositivos computacionais para armazenar, proteger, processar, recuperar e transmitir informações. Pela elevada quantidade de informações armazenadas em formato eletrônico, surge a necessidade de analisar estes dados, extrair conhecimento implícito e utilizá-lo para algum benefício. Este processo é chamado Descoberta de Conhecimento em Bancos de Dados (DCBD), porém utiliza-se na literatura o termo KDD (*Knowledge Discovery in Databases*).

Na Seção 2.2, o conceito de Inteligência Artificial é apresentado. Posteriormente, na Seção 2.3, é enunciado o Aprendizado de Máquina como sub-área da Inteligência Artificial, apresentando seus modos e paradigmas de aprendizado. Já na Seção 2.4 foram apresentadas as etapas que compõem o KDD, desde a definição do problema até a consolidação do conhecimento descoberto. Finalmente na Seção 2.5 foram feitas as considerações finais do capítulo.

2.2 Inteligência Artificial

Inteligência é um conceito primitivo, que ainda não possui uma definição amplamente aceita. Porém, considera-se inteligência a capacidade de resolver problemas. Marvin Minsky,

um dos mais respeitados pesquisadores em Inteligência Artificial, afirmou: "*Nossas mentes contêm processos que nos capacitam a solucionar problemas que consideramos difíceis. Inteligência é o nome que damos a qualquer um destes processos que ainda não compreendemos*" (Minsky, 1985).

Apesar da grande importância das máquinas, elas não são dotadas de inteligência. Ou seja, não são capazes de aprender, reagir adaptativamente ou tomar decisões autônomas. Pelo fato de todas suas ações serem pré-programadas pelo homem, a máquina não sabe reagir a situações desconhecidas. Turing, em seu *imitation game* (Turing, 1950), mostrou que as máquinas não possuem nenhuma inteligência ao serem comparadas com o homem.

Para dar maior capacidade a sistemas computacionais, duas estratégias podem ser utilizadas:

- introduzir no sistema a inteligência, ou melhor, o conhecimento humano;
- fazer o sistema extrair conhecimento implícito.

O objetivo da Inteligência Artificial (IA) é o desenvolvimento de paradigmas ou algoritmos para que as máquinas realizem tarefas cognitivas humanas (Sage, 1990). Ou seja, IA compreende métodos, ferramentas e sistemas para a modelagem de situações que normalmente requerem inteligência humana (Russel e Norvig, 1995). Algumas dessas situações são: tarefas que exigem raciocínio, como planejamento e estratégia; ações de percepção, como reconhecimento de sons e/ou imagens; identificação de grupos distintos ou classes dentro de um conjunto de exemplos; controle de sistemas dinâmicos; entre outras. Para executar estas ações, um sistema de IA deve ser capaz de: armazenar conhecimento, aplicar o conhecimento armazenado para resolver problemas e adquirir novo conhecimento através da experiência (Sage, 1990).

2.3 Aprendizado de Máquina

O Aprendizado de Máquina é uma sub-área da IA que pesquisa métodos computacionais relacionados à aquisição de novos conhecimentos, novas habilidades e novas formas de organizar o conhecimento já existente (Monard et al., 1997). As técnicas de Aprendizado de Máquina possuem duas fases bem definidas: o treinamento e o teste. No treinamento, utiliza-se um conjunto de exemplos, chamado de *conjunto de treinamento*, para aprender o comportamento de um dado sistema. Na fase de teste, um outro conjunto de exemplos, chamado de *conjunto de teste*, é utilizado para validar o treinamento. Ou seja, no teste é avaliado se o treinamento foi bem sucedido ou não.

As técnicas de Aprendizado de Máquina podem ser classificadas pelos modos e paradigmas de aprendizado:

- Modos de Aprendizado

  - Supervisionado: Os exemplos do conjunto de treinamento possuem características individuais e uma classificação ou decisão, permitindo que os mesmos sejam agrupados em classes de exemplos comuns. O treinamento supervisionado procura identificar quais características dos exemplos levam a cada classificação ou decisão. Portanto, este treinamento é guiado pelas classificações ou decisões constantes nos dados.
  - Não-Supervisionado: Os exemplos do conjunto de treinamento possuem apenas suas as características individuais. Sendo assim, o treinamento não-supervisionado procura reconhecer agrupamentos de exemplos comuns ou identificar o perfil dos exemplos sem dispor previamente de nenhuma classificação ou informação decisória.

- Paradigmas de Aprendizado

  - Simbólico: Utilização de expressões lógicas ou regras para representar os exemplos que formam um conceito, um universo. Algumas aplicações do paradigma simbólico são: reconhecimento de padrão e sistemas especialistas.
  - Conexionista: Construções matemáticas inspiradas no modelo biológico do sistema nervoso. Sua representação envolve unidades de processo interconectadas. A aplicação típica do paradigma conexionista são as redes neurais artificiais.
  - Evolucionista: Possui uma analogia direta com a teoria de Darwin, onde sobrevivem os mais bem adaptados ao ambiente. A partir de uma "população" de soluções para um problema (normalmente de busca), avalia-se iterativamente cada solução por alguma função custo, eliminando as piores e proliferando as melhores. Ao final, a melhor dentre as soluções remanescentes é escolhida. Algumas aplicações do paradigma evolucionista são: algoritmos evolucionários, vida artificial.

Este trabalho trata mais especificamente do Aprendizado de Máquina supervisionado, baseado no paradigma simbólico.

2.4 KDD

O processo de descoberta de conhecimento em banco de dados ganha maior apelo a cada dia. Qualquer empresa ou instituição governamental possui informações armazenadas digitalmente, sejam elas em banco de dados, planilhas, documentos de texto, memorandos, etc. Com o advento de computadores com processadores mais poderosos a baixo custo, associado ao avanço dos estudos em organização e mineração de dados, levaram o conceito de KDD (Piatetsky-Shapiro, 1991) a transpor a barreira entre teoria e aplicação. É comum empresas comerciais utilizarem o KDD para traçar estratégias de mercado ou entender o comportamento de seus clientes.

O KDD compreende conceitos de áreas como: Bancos de Dados, IA e Estatística. A Figura 2.1 é clássica na literatura de KDD (Fayyad et al., 1996), apresentando as etapas que compõem todo o processo, as quais são detalhadas nas subseções seguintes.

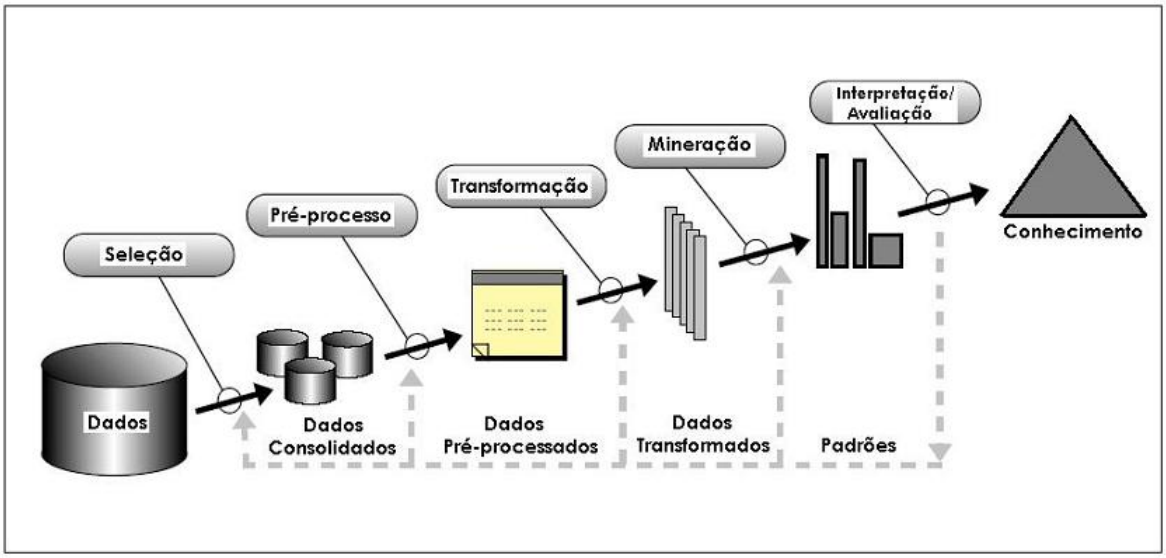

Figura 2.1: Diagrama KDD de Fayyad

2.4.1 Definição do Problema

Todo processo realizado em etapas deve começar pela definição do problema. A partir de um objetivo a ser atingido, encontra-se uma metodologia viável para alcançá-lo. A melhor metodologia tende a ser a que resolve o problema à menor custo computacional, temporal e financeiro.

O passo inicial para o processo é saber exatamente quais são os objetivos finais, ou seja, quais saídas representam sucesso (Noonan, 2000). As etapas seguintes devem ser pensadas para se ter uma previsão de quais resultados parciais poderão ser alcançados.

2.4.2 Seleção dos Atributos Relevantes

Após a definição do problema, deve-se identificar quais serão os dados utilizados em todo o processo. Esses dados podem estar armazenados em bancos de dados, planilhas, documentos de texto, entre outros formatos. Todas as informações relevantes devem ser integradas em um único banco de dados. Um *data warehouse* (Inmon, 1995) viabiliza essa centralização, independente de como estão armazenadas fisicamente as informações.

Estando bem definido o banco de dados a ser utilizado, é feita a seleção dos atributos relevantes, uma etapa de fundamental importância. Um banco de dados pode apresentar:

- Atributos estáticos: possui um determinado valor para cada exemplo, porém este valor não se altera com o passar do tempo. Por exemplo, a data de nascimento de um cliente.

- Atributos dinâmicos: o valor deste atributo pode mudar periodicamente. Por exemplo, consumo de energia mensal de um cliente.

Sendo assim, um atributo estático informa uma característica constante no domínio do tempo, algo que está associado ao exemplo enquanto o mesmo fizer parte do banco de dados. Já um atributo dinâmico expressa variações de uma características do exemplo no domínio do tempo, sendo um atributo de fundamental importância para análises de comportamento.

Quanto mais atributos o banco de dados possui, mais informações sobre os exemplos ele pode conter. Porém, no contexto de KDD, quantidade não significa necessariamente qualidade. A maioria dos sistemas de Aprendizado de Máquina, computacionalmente viáveis, não funcionam bem com uma grande quantidade de atributos (Kira e Rendell, 1992). Existem alguns métodos capazes de encontrar atributos seguindo algum critério de relevância (Caruana e Freitag, 1994). Estudos sobre a seleção de atributos relevantes para Aprendizado de Máquina mostram que somente esta tarefa já é bastante complexa (Pila, 2001).

A participação de especialistas do sistema representado pelo banco de dados é importante nesta etapa, pois os mesmos têm maior conhecimento prático do comportamento dos exemplos, podendo indicar os atributos de maior importância.

Das análises feitas na Subseção 2.4.2, aqueles atributos que não estão relacionados com o objetivo que pretende-se alcançar podem ser descartados. Da mesma forma, aqueles que representam diretamente o sistema em questão são admitidos como relevantes.

2.4.3 Limpeza e Pré-Tratamento dos Dados

De posse do banco de dados e definido os atributos relevantes, inicia-se a limpeza e/ou pré-tratamento dos dados. Esta etapa despende o maior tempo de todo o processo de KDD, cerca de 80% (Manilla, 1994). Contribuem para isso os seguintes fatores:

- atributos encontrados em tabelas distintas do banco de dados e sem a existência de chaves para relacioná-los;
- atributos em branco para determinados exemplos, ou mesmo valores incoerentes para o atributo;
- exemplos duplicados (replicados);
- eventual necessidade de discretização (ou mesmo categorização) de atributos com valores contínuos;
- conversão entre tipos de atributo, por exemplo, de inteiro (200211) para data (11/2002), ou de cadeia de caracteres (“10”) para inteiro (10);
- substituição de determinados valores de atributos por outros pré-estabelecidos;

- remoção de exemplos que são considerados ruídos, ou seja, exemplos que não representam o sistema e que dificultarão o aprendizado;
- obtenção de amostras aleatórias e representativas quando o banco de dados possui uma quantidade muito grande de exemplos;

2.4.4 Transformação dos Dados

Eventualmente, novos atributos podem ser criados à partir dos já existentes, incrementando a quantidade de dados de cada exemplo. Atributos que representam o comportamento dos exemplos muitas vezes necessitam ser criados, como por exemplo: média; desvio-padrão; máximo e mínimo; somatório e produtório; etc.

Após a transformação, os dados são divididos em conjunto de treinamento e conjunto de teste, como dito na Seção 2.3. Somente o conjunto de treinamento será submetido a etapa seguinte.

2.4.5 Mineração

Muitos trabalhos acabam utilizando o nome mineração (ou *data mining*) para denominar todo o processo de KDD. Porém, conceitualmente, esta denominação é equivocada. Fayyad define KDD como todo o processo de descoberta de conhecimento útil em banco de dados (Fayyad et al., 1996). Mineração refere-se a uma única etapa do KDD, onde são aplicados algoritmos específicos de extração de padrões em dados.

Existem muitas técnicas e algoritmos já utilizados para a mineração de dados: ID3 (Quinlan, 1990), C4.5 (Quinlan, 1987) e CART (Breiman et al., 1993), ambos baseados em árvores de decisão; Conjuntos Nebulosos (mais conhecidos na literatura por *Fuzzy Sets*) (Zadeh, 1994); Redes Neurais Artificiais (Haykin, 1998); Mapas Auto-Organizáveis (Kohonen, 1995); Algoritmos Genéticos (Goldberg, 1989); Análise por Grupos (Bolton e Hand, 2001); etc. No trabalho (Mitra et al., 2002) encontra-se uma revisão objetiva e clara das principais técnicas para mineração de dados, apresentando as classes de problemas adequadas para cada técnica.

A teoria de Rough Sets apresenta conceitos que se enquadram no contexto de KDD e que podem ser implementados como algoritmos de mineração (Ziarko e Shan, 1994).

O etapa de mineração tem como saída um conjunto de padrões. Tais padrões podem ser entendidos como vários perfis ou modelos aos quais os exemplos se encaixam ou se assemelham. Geralmente os padrões estão representados simbolicamente na forma de regras *se/então*.

2.4.6 Teste e Análise

O número de padrões extraídos na etapa de mineração depende da semelhança entre os exemplos do conjunto de treinamento. No pior caso, quando não há padrão de comportamento,

o número de padrões é igual ao número de exemplos de treinamento. Normalmente, cada padrão é visto como uma regra se/então, contendo condições e decisões.

Nesta etapa, o grupo de condições de cada regra é comparado ao grupo de condições de cada exemplo do conjunto de teste. Quando há igualdade de condições, verifica-se a decisão da regra e a decisão pré-concebida do exemplo de teste. Ao final, é possível avaliar quais exemplos de teste foram classificados (corretamente ou não) e a qualidade de cada regra na tarefa de classificação ou decisão.

A seleção das regras ou padrões satisfatórios depende do acerto esperado para cada regra, de acordo com a métrica de qualidade de regras empregada.

2.4.7 Consolidação do Conhecimento

Os padrões (regras) extraídos do banco de dados que tiveram sucesso no teste, de acordo com o acerto esperado, representam o conhecimento contido nos dados. Para consolidar o conhecimento, os padrões podem sem interpretados e aplicados individualmente ou serem organizados em um *bancos de regras*, funcionando como um sistema de classificação ou tomada de decisão.

2.5 Considerações Finais

Neste capítulo são enunciadas as definições de IA, Aprendizado de Máquina e KDD, indicando a área de concentração deste trabalho. Todas as etapas do KDD relatadas são revistas em detalhes nos Capítulos 5 e 6.

O próximo capítulo, porém, consiste de uma abordagem prática da teoria de Rough Sets, a qual é utilizada ao longo do trabalho na definição de uma metodologia para detecção de fraudes em consumidores de energia elétrica.

CAPÍTULO

3

Rough Sets - Abordagem Prática

3.1 Introdução

A capacidade de observar certa quantidade de informações (ou dados) e formar um conhecimento é inerente ao ser humano e sua capacidade de aprendizado. A realização desta tarefa pode ser complexa, principalmente quando as informações são desorganizadas, incompletas ou ainda contem partes irrelevantes. A teoria de Rough Sets pode ser utilizada para diminuir as dificuldades na transformação automática de dados em conhecimento.

Do ponto de vista matemático, os conceitos de Rough Sets são simples, envolvendo conjuntos finitos, relações e classes de equivalência. Partindo do princípio de que o mundo real não é exato ou preciso (*crisp*), dados colhidos do mesmo podem ser indiscerníveis ou incertos (*rough*). Rough Sets procura contornar estas incertezas em dados utilizando como fundamento principal a *relação de indiscernibilidade* entre os exemplos de um banco de dados. Esta relação está fortemente associada aos valores dos atributos que compõem este banco de dados, o qual será redefinido posteriormente para melhor representar os repositórios de dados.

Os conceitos de Rough Sets e Fuzzy Sets possuem enfoques distintos: Rough Sets tratam da indiscernibilidade e da incerteza em dados, enquanto Fuzzy Sets consideram a forma como são manipulados os valores contidos nos dados, independente da indiscernibilidade (Dubois e Prade, 1990).

A Seção 3.2 lista algumas aplicações da teoria de Rough Sets, a qual é apresentada na Seção 3.3, englobando: a representação de dados, os redutos, os *conceitos*, as aproximações de *conceitos* e a discretização. Finalmente, na Seção 3.4, são feitas as considerações finais deste capítulo.

3.2 Aplicações

Apesar do curto período de existência, Rough Sets vem sendo aplicado em diversas áreas (Polkowski et al., 1998):

- Aprendizado de regras de decisão;
- Mineração de dados;
- Processamento de sinais (som e imagem);
- Sistemas especialistas e de suporte à decisão;
- Seleção de atributos relevantes;
- Filtragem de sinais;
- Agrupamento (*clustering*);

3.3 Teoria de Rough Sets

A teoria de Rough Sets foi proposta em 1982 por Zdzislaw Pawlak (Pawlak, 1982). Posteriormente, em 1991, Pawlak publicou o livro "*Rough Sets: Theoretical Aspects of Reasoning about Data*" (Pawlak, 1991), constando a fundamentação teórica e mostrando algumas aplicações.

3.3.1 Representação dos Dados

As informações colhidas do mundo real podem ser organizadas em banco de dados. Essa estrutura pode ser simplificada para um única tabela, chamada de *Sistema de Informação*[1] (SI). Na Tabela 3.1 é mostrado um exemplo de SI (Pawlak et al., 1995).

As linhas da Tabela 3.1 representam os exemplos, objetos, registros ou casos (Clientes). As colunas são os atributos condicionais (Tipo de Ligação, Classe, Média de Consumo Anual) e de decisão (Fraudador) para cada exemplo. Pode-se ter vários atributos condicionais e vários atributos de decisão, embora estes últimos apareçam geralmente como um único atributo, como na Tabela 3.1. Qualquer sistema baseado em casos pode ser representado por um SI, onde nas linhas encontram-se os exemplos e nas colunas os atributos.

Formalmente, um SI é definido por $\mathcal{A} = (U, A \cup D)$, em que:

[1]Para simplificar a nomenclatura da representação de dados, foi adotado o termo Sistema de Informação (SI). No Capítulo 4, um SI é redefinido por conveniência à teoria.

Tabela 3.1: Sistema de Informação

	Atributos			**Decisão**
Cliente	Tipo de Ligação	Classe	Média de Consumo Anual	Fraudador
e1	1	1	Normal	Não
e2	1	1	Alta	Sim
e3	1	1	Baixa	Sim
e4	2	1	Normal	Não
e5	2	2	Alta	Não
e6	2	1	Baixa	Sim

- U: corresponde ao conjunto contendo todos os exemplos. Para a Tabela 3.1, $U = \{e1, e2, e3, e4, e5, e6\}$.
- A: corresponde ao conjunto contendo todos os atributos condicionais. Para a Tabela 3.1, $A = \{TipodeLigacao, Classe, MediadeConsumoAnual\}$. Cada atributo condicional pode ser representado por a.
- D: corresponde ao conjunto contendo todos os atributos de decisão. Para a Tabela 3.1, $D = \{Fraudador\}$. Cada atributo condicional pode ser representado por d.

3.3.2 Redutos

Considerando o conjunto A da Tabela 3.1, observa-se que o mesmo consegue distinguir todos elementos em U, ou seja, todos os exemplos $e1$, $e2$, $e3$, $e4$, $e5$ e $e6$ são discerníveis entre si, são *elementares*. Considerando agora o subconjunto $\{TipodeLigacao, Classe\} \subset A$, o conjunto U é particionado em subconjuntos não-elementares: $\{e1, e2, e3\}$, $\{e4, e6\}$ e $\{e5\}$. Sendo assim, somente os atributos $TipodeLigacao$ e $Classe$ não conseguem discernir todos os exemplos, i.e., não conseguem fazer a função de A. Porém, o subconjunto de atributos $\{TipodeLigacao, MediadeConsumoAnual\} \subset A$ consegue particionar U em subconjuntos elementares. Isto é, somente os atributos $TipodeLigacao$ e $MediadeConsumoAnual$ podem distinguir todos elementos de U. Conclui-se então que o atributo $Classe$ é *redundante*. O conjunto $P = \{TipodeLigacao, MediadeConsumoAnual\}$ não possui atributos redundantes e é chamado de *reduto* do conjunto A.

Formalmente, um conjunto de atributos P é reduto (ou cobertura) de A se $P \subseteq A$ mantém as relações de indiscernibilidade de A. Em outras palavras, se P tem cardinalidade menor ou igual a A e consegue manter a mesma representação dos exemplos de um dado SI, então P é um reduto de A.

Baseado no reduto P encontrado para o conjunto A da Tabela 3.1, um novo SI é mostrado na Tabela 3.2.

Tabela 3.2: Sistema de Informação reduzido

	Atributos		**Decisão**
Cliente	Tipo de Ligação	Média de Consumo Anual	Fraudador
e1	1	Normal	Não
e2	1	Alta	Sim
e3	1	Baixa	Sim
e4	2	Normal	Não
e5	2	Alta	Não
e6	2	Baixa	Sim

Apesar da Tabela 3.2 ilustrar um reduto para o SI da Tabela 3.1, redutos não são necessariamente únicos. De acordo com a dependência entre os atributos (Pawlak, 1991), pode existir mais de um único reduto para um dado conjunto de atributos.

Com a diminuição do número de atributos, menos dados são necessários para representar o mesmo conhecimento. Essa redução é ainda mais relevante quando o SI possui muitos atributos linearmente dependentes. Na verdade, encontrar um reduto é encontrar os atributos linearmente independentes de um determinado sistema, representado por um SI.

O algoritmo que encontra o reduto mínimo tem complexidade computacional *NP-difícil*, necessitando uma alocação de memória na ordem de $O(kn^2/2)$, sendo k o número de atributos e n o número de exemplos (Pawlak, 1991). Existem heurísticas (Hu et al., 2003) que conseguem encontrar redutos com um menor custo computacional, porém não garantem que os redutos sejam mínimos (ótimos), i.e., tenham a menor cardinalidade possível.

3.3.3 Conceitos

Além dos atributos condicionais analisados para a busca de redutos em SI, os atributos de decisão também desempenham um importante papel na teoria de Rough Sets.

Considerando o conjunto D dos atributos de decisão da Tabela 3.2 ($\{Fraudador\}$), o mesmo divide o conjunto U em dois subconjuntos: $\{e1, e4, e5\}$ e $\{e2, e3, e6\}$. Cada subconjunto é chamado de *conceito*. O primeiro *conceito* corresponde aos exemplos não-fraudadores, enquanto que o segundo abrange os exemplos fraudadores. Os *conceitos* determinam as classes nas quais os exemplos se encontram.

A partir dos atributos condicionais em A, pode-se determinar a que *conceito* (ou classe) um dado exemplo se enquadra. Chama-se esse tipo de tarefa de classificação. Dada a Tabela 3.2, um conjunto de *regras de classificação* pode ser gerado:

1. Se $MediadeConsumoAnual =$ Normal $\rightarrow Fraudador =$ Não
2. Se $TipodeLigacao = 2$ e $MediadeConsumoAnual =$ Alta $\rightarrow Fraudador =$ Não

3. Se $TipodeLigacao = 1$ e $MediadeConsumoAnual =$ Alta $\rightarrow Fraudador =$ Sim

4. Se $MediadeConsumoAnual =$ Baixa $\rightarrow Fraudador =$Sim

Cada linha da Tabela 3.2 deu origem a uma regra distinta. As regras que foram originadas pelos exemplos $e1$ e $e4$, como também por $e3$ e $e6$, foram simplificadas e reduzidas nas regras 1 e 4, respectivamente. Sendo assim, o banco de regras obtido classifica todos exemplos da Tabela 3.2. Porém, nem sempre este método pode ser usado diretamente. Para demonstrar uma situação freqüente e problemática, considere a Tabela 3.3 construída com o incremento dos elementos $e7$ e $e8$ à Tabela 3.2.

Tabela 3.3: Sistema de Informação inconsistente

	Atributos		**Decisão**
Cliente	Tipo de Ligação	Média de Consumo Anual	Fraudador
e1	1	Normal	Não
e2	1	Alta	Sim
e3	1	Baixa	Sim
e4	2	Normal	Não
e5	2	Alta	Não
e6	2	Baixa	Sim
e7	2	Alta	Sim
e8	2	Baixa	Não

Os *conceitos* definidos pelo atributo $\{Fraudador\}$ da Tabela 3.3 são $N = \{e1, e4, e5, e8\}$ e $F = \{e2, e3, e6, e7\}$. Porém, os exemplos $e5$ e $e7$, apesar de possuírem os mesmos valores para os atributos condicionais, pertencem a diferentes *conceitos*. O mesmo ocorre entre os exemplos $e6$ e $e8$. Essas inconsistências na Tabela 3.3 impedem a criação de duas regras:

1. Se $TipodeLigacao = 2$ e $MediadeConsumoAnual =$ Alta $\rightarrow Fraudador =$?

2. Se $TipodeLigacao = 2$ e $MediadeConsumoAnual =$ Baixa $\rightarrow Fraudador =$?

Para tratar essas situações, em que não é possível definir quais serão as classificações das regras, a teoria de Rough Sets define três subconjuntos de U.

3.3.4 Aproximação Inferior, Superior e Região de Fronteira

Considerando X como um dos *conceitos* de um SI, pode-se encontrar um subconjunto de X com elementos que **com certeza** estão contidos no *conceito* X. Este subconjunto chama-se Aproximação Inferior de X, ou simplesmente $\underline{A}X$, sendo A o conjunto de atributos condicionais considerados. Para a tabela 3.3, se $N = \{e1, e4, e5, e8\}$, então $\underline{A}N = \{e1, e4\}$. Da

mesma forma, se $F = \{e2, e3, e6, e7\}$, então $\underline{A}F = \{e2, e3\}$. Nota-se que $\underline{A}N$ será sempre um subconjunto de N, ou seja, $\underline{A}N \subseteq N$ (ocorrendo o mesmo para $\underline{A}F$).

A Aproximação Superior de X, ou simplesmente $\overline{A}X$, corresponde a um subconjunto de U com elementos que **podem** estar contidos em um *conceito* X. Para a tabela 3.3, se $N = \{e1, e4, e5, e8\}$, então $\overline{A}N = \{e1, e4, e5, e6, e7, e8\}$. Da mesma forma, se $F = \{e2, e3, e6, e7\}$, então $\overline{A}F = \{e2, e3, e5, e6, e7, e8\}$. Nota-se que $\overline{A}N$ será sempre um subconjunto de U e conterá todo conjunto N, ou seja, $N \subseteq \overline{A}N \subseteq U$ (ocorrendo o mesmo para $\overline{A}F$).

A Região de Fronteira de X, ou simplesmente $BN_A(X)$, corresponde a um subconjunto de U com elementos que pertencem a $\overline{A}X$ mas não pertencem a $\underline{A}X$, ou seja, $BN_A(X) = \overline{A}X - \underline{A}X$. Se $BN_A(X) = \emptyset$, então $\overline{A}X$ e $\underline{A}X$ são os mesmos conjuntos, i.e., o SI não possui exemplos indiscerníveis. Conseqüentemente, quanto maior a cardinalidade de $BN_A(X)$, maior a indiscernibilidade entre os *conceitos*.

A Figura 3.1 ajuda a compreender a distribuição dos exemplos da Tabela 3.3 dentro dos *conceitos*. Para o *conceito* $F = \{e2, e3, e6, e7\}$ de fraudador, os exemplos que com certeza são fraudadores estão no bloco preto ($\underline{A}F$). Os exemplos que podem ser fraudadores estão nos blocos preto, cinza e cinza claro ($\overline{A}F$). Já os exemplos que com certeza não são fraudadores estão no bloco branco ($U - \overline{A}X$). A mesma análise pode ser feita para o *conceito* $N = \{e1, e4, e5, e8\}$ de não fraudador.

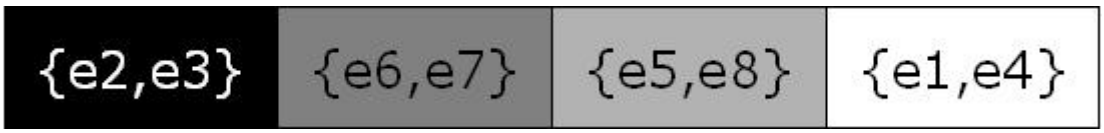

Figura 3.1: Distribuição dos exemplos da Tabela 3.3 com relação aos *conceitos*.

Com os exemplos de um SI contidos nos conjuntos $\underline{A}X$, $\overline{A}X$ e $BN_A(X)$, os mesmos ficam organizados de acordo com suas pertinências aos *conceitos*, eliminando, de certa forma, as inconsistências ou indiscernibilidades. Caso deseje-se encontrar os exemplos que com certeza são fraudadores, basta determinar $\underline{A}X$. Quando a certeza não é obrigatória e deseja-se determinar os possíveis fraudadores, encontra-se $\overline{A}X$. Em uma análise dos exemplos aos quais não se pode ter certeza a que *conceito* pertencem, avalia-se $BN_A(X)$.

Qualidade das Aproximações

Dado o SI, pode-se avaliar a qualidade das aproximações encontradas em função das cardinalidades dos conjuntos U, $\underline{A}X$ e $\overline{A}X$. A qualidade da aproximação inferior $\alpha(\underline{A}X)$ corresponde ao percentual de elementos que com certeza pertencem ao *conceito* X:

$$\alpha(\underline{A}X) = \frac{|\underline{A}X|}{|U|}$$

Para os fraudadores da Tabela 3.3:

$$\alpha(\underline{A}F) = \frac{|\{e2, e3\}|}{|\{e1, e2, e3, e4, e5, e6, e7, e8\}|} = 0.25$$

A qualidade da aproximação superior $\alpha(\overline{A}X)$ corresponde ao percentual de elementos que possivelmente pertençam ao *conceito* X:

$$\alpha(\overline{A}X) = \frac{|\overline{A}X|}{|U|}$$

Para os fraudadores da Tabela 3.3:

$$\alpha(\overline{A}F) = \frac{|\{e2, e3, e5, e6, e7, e8\}|}{|\{e1, e2, e3, e4, e5, e6, e7, e8\}|} = 0.75$$

Já o coeficiente de incerteza corresponde à qualidade da aproximação dos *conceitos*. Se $\alpha(X) = 1$, o *conceito* X é preciso (*crisp*). Se $0 < \alpha(X) < 1$, o *conceito* X é parcialmente impreciso (*rough*). Se $\alpha(X) = 0$, o *conceito* X é totalmente impreciso (*rough*):

$$\alpha(X) = \frac{|\underline{A}X|}{|\overline{A}X|}$$

Para os fraudadores da Tabela 3.3:

$$\alpha(F) = \frac{|\{e2, e3\}|}{|\{e2, e3, e5, e6, e7, e8\}|} = 0.33$$

3.3.5 Discretização

A operação base dos conceitos de Rough Sets é a comparação iterativa dos valores dos atributos de cada exemplo. Quando o SI apresenta atributos que podem admitir muitos valores, ou mesmo atributos contínuos, há a necessidade de discretização.

A discretização de atributos pode manter ou mesmo modificar as relações de indiscernibilidade entre os exemplos de um SI. Isto porque este processo pode simplesmente trocar os valores de atributos contínuos por valores discretos, tornando finito o número de possíveis valores, sem modificar a indiscernibilidade entre os exemplos. Ou, então, pode-se realizar uma discretização que tornam indiscerníveis os exemplos com valores de atributos muito próximos, promovendo o aumento da região de fronteira entre classes de decisão de um SI.

Os intervalos (ou faixas) que determinam a discretização de um atributo podem ser definidos explicitamente ou mesmo por algoritmos de discretização. Estes algoritmos são de alta complexidade computacional (NP-completo ou NP-difícil), podendo esta complexidade crescer exponencialmente com o número de atributos a serem discretizados (Komorowski et al., 1999). Como a discretização é uma etapa necessária não somente em Rough Sets, existem heurísticas eficientes para a discretização de atributos com valores contínuos (reais) baseadas em

Rough Sets, Aprendizado de Máquina, Reconhecimento de Padrão e KDD (Lenarcik e Piasta, 1992) (Lenarcik e Piasta, 1993) (Nguyen e Skowron, 1995) (Lenarcik e Piasta, 1997) (Nguyen, 1997) (Chmielewski e Grzymala-Busse, 1994) (Dougherty et al., 1995) (Fayyad e Irani, 1992) (Murthy et al., 1993).

Algoritmo Básico de Discretização usando Rough Sets e Lógica Booleana

Dado um SI definido por $\mathcal{A} = (U, A \cup \{d\})$, em que $V_a = [v_a, w_a)$ representa o intervalo real de possíveis valores de a, deseja-se encontrar uma partição P_a de V_a para qualquer $a \in A$. Qualquer partição de V_a é definida por uma seqüência de *cortes* $v_1 < v_2 < ... < v_k$ que definem as faixas de discretização de a. Sendo assim, este processo de discretização consiste em encontrar um conjunto de cortes que satisfaça as condições iniciais do SI, ou seja, mantenha as relações de indiscernibilidade.

Para exemplificar este processo, considere o SI da Tabela 3.4, o qual apresenta dois atributos condicionais contínuos e um atributo de decisão.

Tabela 3.4: Sistema de Informação a ser discretizado.

U	x	y	d
$u1$	0.6	1.6	1
$u2$	1.4	2	0
$u3$	1.4	2.4	1
$u4$	1.6	3	0
$u5$	1.8	1	0
$u6$	1.8	2.4	1
$u7$	2.6	1	1
$u8$	2.6	3	0

O conjunto de possíveis valores são $V_x = [0, 3)$ e $V_y = [0, 4)$. O conjunto de valores apresentados pelos exemplos são $x(U) = \{0.6, 1.4, 1.6, 1.8, 2.6\}$ e $y(U) = \{1, 1.6, 2, 2.4, 3\}$. Os intervalos entre os valores dos atributos determinam os cortes iniciais, que são os pontos-médios entre os valores. Cada corte é formalmente representado por (a, c), em que $c \in V_a$. A Tabela 3.5 ilustra os intervalos e respectivos cortes da Tabela 3.4:

Um conjunto de cortes P define novos atributos condicionais a^P, os quais formam um novo SI discretizado. Por exemplo, a partir de $P = \{(x, 1), (x, 2.2), (y, 2.2), (y, 2.7)\}$ e do SI da Tabela 3.4, define-se x^P e y^P como ilustrado na Tabela 3.6. Valores de x menores que 1 foram preenchidos com 0, entre $[1, 2.2)$ com 1 e entre $[2.2, 3)$ com 2. O mesmo procedimento foi realizado para y. É fácil ver que o novo SI discretizado manteve todas as relações de indiscernibilidade entre os exemplos, com um número reduzido de valores em seus atributos.

Tabela 3.5: Intervalos e respectivos cortes.

intervalo	corte	intevalo	corte
$[0.6, 1.4)$	$(x, 1)$	$[1, 1.6)$	$(y, 1.3)$
$[1.4, 1.6)$	$(x, 1.5)$	$[1.6, 2)$	$(y, 1.8)$
$[1.6, 1.8)$	$(x, 1.7)$	$[2, 2.4)$	$(y, 2.2)$
$[1.8, 2.6)$	$(x, 2.2)$	$[2.4, 3)$	$(y, 2.7)$

Tabela 3.6: Sistema de Informação discretizado.

U^P	x^P	y^P	d
$u1$	0	0	1
$u2$	1	0	0
$u3$	1	1	1
$u4$	1	2	0
$u5$	1	0	0
$u6$	1	1	1
$u7$	2	0	1
$u8$	2	2	0

A questão central da discretização baseada em Rough Sets e Lógica Booleana é como encontrar um conjunto P ótimo (com um número mínimo de elementos) de tal sorte que o SI discretizado mantenha as relações de indiscernibilidade entre os exemplos.

O primeiro passo para encontrar um conjunto P ótimo é transformar cada corte (e seu respectivo intervalo) em uma variável booleana. Considerando os cortes da Tabela 3.5, define-se o conjunto de variáveis booleanas $VB(A) = \{p_1^x, p_2^x, p_3^x, p_4^x, p_1^y, p_2^y, p_3^y, p_4^y\}$, em que p_1^x equivale à $[0.6, 1.4)$, p_2^x à $[1.4, 1.6)$, p_1^y à $[1, 1.6)$ e assim sucessivamente.

Para cada par de exemplos com decisão distintas do SI da Tabela 3.4, constrói-se uma matriz de discernimento que é preenchida com elementos de $VB(A)$ conforme ilustra a Tabela 3.7.

Tabela 3.7: Matriz de discernimento.

	$u2$	$u4$	$u5$	$u8$
$u1$	$\{p_1^x, p_2^y\}$	$\{p_1^x, p_2^x, p_2^y, p_3^y, p_4^y\}$	$\{p_1^x, p_2^x, p_3^x, p_1^y\}$	$\{p_1^x, p_2^x, p_3^x, p_4^x, p_2^y, p_3^y, p_4^y\}$
$u3$	$\{p_3^y\}$	$\{p_2^x, p_4^y\}$	$\{p_2^x, p_3^x, p_1^y, p_2^y, p_3^y\}$	$\{p_2^x, p_3^x, p_4^x, p_4^y\}$
$u6$	$\{p_2^x, p_3^x, p_3^y\}$	$\{p_3^x, p_4^y\}$	$\{p_1^y, p_2^y, p_3^y\}$	$\{p_4^x, p_4^y\}$
$u7$	$\{p_2^x, p_3^x, p_4^x, p_1^y, p_2^y\}$	$\{p_3^x, p_4^x, p_1^y, p_2^y, p_3^y, p_4^y\}$	$\{p_4^x\}$	$\{p_1^y, p_2^y, p_3^y, p_4^y\}$

O conteúdo das células da matriz de discernimento representam as variáveis booleanas contidas no intervalo determinado pelos valores de atributos do par de exemplos. Considerando o

par ($u1$,$u2$), por exemplo, a variável booleana p_1^x está contida no intervalo $[0.6, 1.4]$ de x e p_2^y está contida no intervalo $[1.6, 2]$ de y. O mesmo procedimento é realizado no preenchimento das demais células da matriz.

As variáveis booleanas contidas na célula de um par de exemplos são aquelas que, caso tornem-se cortes de P, conseguirão distinguir estes mesmos exemplos. Sendo assim, deriva-se da matriz de discernimento um expressão booleana formada pela conjunção das disjunções das variáveis booleanas de cada célula, como mostrada abaixo:

$$\Phi^{\mathcal{A}} = (p_1^x \vee p_2^y) \wedge (p_1^x \vee p_2^x \vee p_2^y \vee p_3^y \vee p_4^y) \wedge (p_1^x \vee p_2^x \vee p_3^x \vee p_1^y)$$
$$\wedge(p_1^x \vee p_2^x \vee p_3^x \vee p_4^x \vee p_2^y \vee p_3^y \vee p_4^y) \wedge p_3^y \wedge (p_2^x \vee p_4^y) \wedge (p_2^x \vee p_3^x \vee p_1^y \vee p_2^y \vee p_3^y)$$
$$\wedge(p_2^x \vee p_3^x \vee p_4^x \vee p_4^y) \wedge (p_2^x \vee p_3^x \vee p_3^y) \wedge (p_3^x \vee p_4^y) \wedge (p_1^y \vee p_2^y \vee p_3^y) \wedge (p_4^x \vee p_4^y)$$
$$\wedge(p_2^x \vee p_3^x \vee p_4^x \vee p_1^y \vee p_2^y) \wedge (p_3^x \vee p_4^x \vee p_1^y \vee p_2^y \vee p_3^y \vee p_4^y) \wedge (p_4^x) \wedge (p_1^y \vee p_2^y \vee p_3^y \vee p_4^y)$$

Simplificando a expressão $\Phi^{\mathcal{A}}$ e passando-a da forma CNF (*Conjunctive Normal Form*) para a forma DNF (*Disjunctive Normal Form*), encontra-se:

$$\Phi^{\mathcal{A}} = (p_1^x \vee p_2^y) \wedge (p_1^x \vee p_2^x \vee p_2^y \vee p_3^y \vee p_4^y) \wedge (p_1^x \vee p_2^x \vee p_3^x \vee p_1^y)$$
$$\wedge(p_1^x \vee p_2^x \vee p_3^x \vee p_4^x \vee p_2^y \vee p_3^y \vee p_4^y) \wedge p_3^y \wedge (p_2^x \vee p_4^y) \wedge (p_2^x \vee p_3^x \vee p_1^y \vee p_2^y \vee p_3^y)$$
$$\wedge(p_2^x \vee p_3^x \vee p_4^x \vee p_4^y) \wedge (p_2^x \vee p_3^x \vee p_3^y) \wedge (p_3^x \vee p_4^y) \wedge (p_1^y \vee p_2^y \vee p_3^y) \wedge (p_4^x \vee p_4^y)$$
$$\wedge(p_2^x \vee p_3^x \vee p_4^x \vee p_1^y \vee p_2^y) \wedge (p_3^x \vee p_4^x \vee p_1^y \vee p_2^y \vee p_3^y \vee p_4^y) \wedge (p_4^x) \wedge (p_1^y \vee p_2^y \vee p_3^y \vee p_4^y)$$

Cada conjunção acima representa um conjunto de cortes P válido. A partir de qualquer destes conjuntos encontra-se um SI discretizado $\Phi^{\mathcal{A}^P}$ que mantém as relações de indiscernibilidade de $\Phi^{\mathcal{A}}$ (Komorowski et al., 1999). Por exemplo, a conjunção $(p_1^x \wedge p_2^x \wedge p_3^y \wedge p_4^y)$ representa o conjunto $P = \{(x, 1), (x, 2.2), (y, 2.2), (y, 2.7)\}$, o qual foi aplicado no SI da Tabela 3.4, gerando o da Tabela 3.6. Como já foi observado, estes SI possuem as mesmas relações de indiscernibilidade entre os exemplos.

Heurísticas de Discretização

Embora o algoritmo de discretização apresentado acima seja eficiente, encontrando os cortes mínimos válidos, o mesmo é NP-difícil (Komorowski et al., 1999). Uma alternativa a esta complexidade são as heurísticas de discretização, dentre elas, a baseada na estratégia de Johnson (Johnson, 1974). Esta heurística é semelhante ao algoritmo básico de discretização descrito acima.

Considerando os pares de exemplos com decisões distintas do SI da Tabela 3.4 e as variáveis booleanas $VB(A) = \{p_1^x, p_2^x, p_3^x, p_4^x, p_1^y, p_2^y, p_3^y, p_4^y\}$ já enunciadas, constrói-se uma tabela booleana auxiliar como mostrada pela Tabela 3.8.

Tabela 3.8: Tabela booleana auxiliar à heurística de discretização.

	p_1^x	p_2^x	p_3^x	p_4^x	p_1^y	p_2^y	p_3^y	p_4^y
$(u1, u2)$	1	0	0	0	0	1	0	0
$(u1, u4)$	1	1	0	0	0	1	1	1
$(u1, u5)$	1	1	1	0	1	0	0	0
$(u1, u8)$	1	1	1	1	0	1	1	1
$(u3, u2)$	0	0	0	0	0	0	1	0
$(u3, u4)$	0	1	0	0	0	0	0	1
$(u3, u5)$	0	1	1	0	1	1	1	0
$(u3, u8)$	0	1	1	1	0	0	0	1
$(u6, u2)$	0	1	1	0	0	0	1	0
$(u6, u4)$	0	0	1	0	0	0	0	1
$(u6, u5)$	0	0	0	0	1	1	1	0
$(u6, u8)$	0	0	0	1	0	0	0	1
$(u7, u2)$	0	1	1	1	1	1	0	0
$(u7, u4)$	0	0	1	1	1	1	1	1
$(u7, u5)$	0	0	0	1	0	0	0	0
$(u7, u8)$	0	0	0	0	1	1	1	1

Nota-se que a Tabela 3.8 é semelhante à matriz de discernimento da Tabela 3.7. Células armazenando o valor 1 simbolizam que o corte relacionado à variável booleana da coluna está no intervalo entre os valores do par de exemplos da linha. Caso o corte não esteja no referido intervalo, a célula recebe o valor 0.

A heurística consiste em encontrar a coluna da Tabela 3.8 com o maior número de valores 1, ou seja, o corte (representado pela variável booleana) que separa o maior número de exemplos com decisões distintas. Encontrada a coluna, remove-se todas as linhas marcadas com 1 na referida coluna, e posteriormente a própria coluna. Repete-se este passo até que a tabela não possua mais nenhuma linha. Os cortes associados às colunas eliminadas serão os cortes em P que levarão a discretização do SI. A Tabela 3.9 ilustra a situação da Tabela 3.8 após as sucessivas eliminações de colunas e linhas.

Analisando a Tabela 3.9, vê-se que a cada busca pela coluna com o maior número de valores 1, pode haver mais de uma opção, ou seja, colunas como o mesmo número máximo de valores 1. O critério para a escolha da coluna será relevante na sucessão da heurística, determinando quais dos seguintes cortes serão selecionados. Na execução da heurística ilustrada na Tabela 3.9, foram selecionadas, seqüencialmente, as variáveis booleanas p_2^x, p_2^y, p_4^x, p_3^x e p_3^y. Se a heurística

tivesse selecionado, por exemplo, as variáveis p_3^y, p_4^x, p_4^y e p_1^x seqüencialmente, teria-se chegado ao mínimo de cortes encontrado pelo algoritmo de discretização apresentado anteriormente.

Embora a heurística seja eficiente, encontrando um conjunto de cortes válidos, a mesma apresenta fatores desfavoráveis: custo computacional $O(kn^3)$ para encontrar cada corte c e alocação de memória na ordem de $O(kn^2)$, sendo k o número de atributos e n o número de exemplos do SI.

3.4 Considerações Finais

Neste capítulo a teoria de Rough Sets foi apresentada de forma prática e objetiva. Os principais conceitos envolvidos em Rough Sets foram levantados através de exemplos e definições claras, contextualizadas ao problema da detecção de fraudes em consumidores de energia elétrica. Foi dada ênfase ao processo de discretização de dados, fundamental para a teoria de Rough Sets.

No próximo capítulo é feita uma abordagem teórica de Rough Sets, permitindo uma compreensão formal e aprofundada.

Tabela 3.9: Estado da tabela booleana auxiliar com a execução da heurística de discretização.

	p_1^x	p_3^x	p_4^x	p_1^y	p_2^y	p_3^y	p_4^y
$(u1, u2)$	1	0	0	0	1	0	0
$(u3, u2)$	0	0	0	0	0	1	0
$(u6, u4)$	0	1	0	0	0	0	1
$(u6, u5)$	0	0	0	1	1	1	0
$(u6, u8)$	0	0	1	0	0	0	1
$(u7, u4)$	0	1	1	1	1	1	1
$(u7, u5)$	0	0	1	0	0	0	0
$(u7, u8)$	0	0	0	1	1	1	1

	p_1^x	p_3^x	p_4^x	p_1^y	p_3^y	p_4^y
$(u3, u2)$	0	0	0	0	1	0
$(u6, u4)$	0	1	0	0	0	1
$(u6, u8)$	0	0	1	0	0	1
$(u7, u5)$	0	0	1	0	0	0

	p_1^x	p_3^x	p_1^y	p_3^y	p_4^y
$(u3, u2)$	0	0	0	1	0
$(u6, u4)$	0	1	0	0	1

	p_1^x	p_1^y	p_3^y	p_4^y
$(u3, u2)$	0	0	1	0

	p_1^x	p_1^y	p_4^y

CAPÍTULO

4

Rough Sets - Abordagem Teórica

4.1 Introdução

Como visto no Capítulo 3, os conceitos de Rough Sets são de fácil compreensão prática e aplicação. A despeito de sua utilização direta como técnica de Inteligência Artificial, Rough Sets possui uma fundamentação teórica bem consolidada. Esta abordagem teórica pode ser encontrada no livro de Zdzislaw Pawlak, intitulado "*Rough Sets: Theoretical Aspects of Reasoning about Data*" (Pawlak, 1991). Este capítulo está baseado nesta referência e procura apresentar, de forma mais didática e contextual, os pontos centrais desta abordagem.

Antes de introduzir os aspectos teóricos de Rough Sets, na Seção 4.2 são enunciadas algumas definições de base. Posteriormente, na Seção 4.3 é apresentada a idéia de conhecimento e na Seção 4.4 a teoria de Rough Sets é finalmente discutida. Objetivando a eliminação de conhecimentos supérfluos, a Seção 4.5 apresenta a idéia de reduto e núcleo. Na Seção 4.6 a forma de representação do conhecimento é definida por Sistemas de Informação. Posteriormente, na Seção 4.7, os Sistemas de Informação ganham um atributo de decisão, alcançando a idéia de Tabelas de Decisão. Finalmente na Seção 4.8 as considerações finais do capítulo são realizadas.

4.2 Objeto e Conhecimento

A teoria de Rough Sets está calcada em dois elementos: **objetos** e o **conhecimento** acerca dos mesmos. Os objetos são instâncias (ou exemplos, registros) de qualquer elemento real ou imaginário. Ou seja, objetos podem representar seres humanos, objetos concretos, medidas de

algum fenômeno amostradas no tempo, ou qualquer outra entidade que se possa imaginar. A um conjunto de objetos, doravante denominado *universo de discurso* (ou simplesmente *universo*), é possível aplicar uma ou mais características, definindo uma classificação de objetos. A estas características dá-se o nome de conhecimento. Portanto, dado um universo e o conhecimento disponível sobre o mesmo, é possível realizar classificações ou partições neste universo. Por exemplo, dado um conjunto de pessoas (universo) e seus respectivos sexos (conhecimento), é possível encontrar uma partição (classificação) deste conjunto: o subconjunto de homens e o subconjunto de mulheres.

4.3 Base de Conhecimento

Dado o conjunto $U \neq \emptyset$ como sendo um universo de objetos, um subconjunto $X \subseteq U$ é chamado de *conceito* (ou *categoria*) em U. Uma família de *conceitos* em U é denominada *conhecimento* sobre U. É desejado que o conhecimento seja $C = \{X_1, X_2, ..., X_n\}$ tal que $X_i \subseteq U, X_i \neq \emptyset, X_i \cap X_j = \emptyset$ para $i \neq j, j = 1, ..., n$ e $\bigcup X_i = U$. Em outras palavras, é interessante que um conhecimento defina uma classificação exata sobre U, ou seja, defina *conceitos* com intersecção vazia.

Pode-se encontrar mais de um conhecimento sobre o universo U, levando os objetos à classificações possivelmente distintas. A esta família de conhecimentos (ou classificações) sobre U dá-se o nome de *base de conhecimento*.

Devido ao formalismo matemático e às facilidades de manipulação, é conveniente redefinir os conceitos apresentados, baseando-os agora em relações e classes de equivalência.

Sendo R uma relação de equivalência sobre U (o mesmo que um conhecimento sobre U), U/R determina a família de classes de equivalência de R. Uma classe de equivalência é um *conceito* em R, representada simbolicamente por $[x]_R$, onde $x \in U$. Uma base de conhecimento é um sistema relacional $K = (U, \mathbf{R})$, onde $U \neq \emptyset$ é um conjunto finito de objetos chamado universo e $\mathbf{R}$ é uma família de relações de equivalência sobre U.

Considerando $\mathbf{P} \subseteq \mathbf{R}$ e sendo $\mathbf{P} \neq \emptyset$, define-se a intersecção de todas as relações de equivalência $\bigcap \mathbf{P}$ como sendo uma **única** relação de equivalência chamada *relação de indiscernibilidade* sobre $\mathbf{P}$, simbolicamente $IND(\mathbf{P})$. As classes de equivalência de $IND(\mathbf{P})$ são subconjuntos não-vazios de U originados das possíveis intersecções entre as classes das relações de equivalência contidas em $\mathbf{P}$, ou seja:

$$[x]_{IND(\mathbf{P})} = \bigcap_{R \in \mathbf{P}} [x]_R$$

Como $IND(\mathbf{P})$ é uma relação de equivalência, $U/IND(\mathbf{P})$ (ou simplesmente $U/\mathbf{P}$) determina também uma família de classes de equivalência sobre U.

Uma relação de equivalência $R \in \mathbf{R}$ qualquer leva à um R-conhecimento elementar sobre U e à classes de equivalência ou R-*conceitos* elementares do conhecimento $\mathbf{R}$. Já uma relação

de equivalência $IND(\mathbf{P})$, onde $\mathbf{P} \neq \emptyset$ e $\mathbf{P} \subseteq \mathbf{R}$, leva à um $\mathbf{P}$-conhecimento básico sobre U e à $\mathbf{P}$-*conceitos* básicos do conhecimento $\mathbf{P}$. A união dos $\mathbf{P}$-*conceitos* básicos é chamada $\mathbf{P}$-*conceitos* e a família de todos *conceitos* em uma base de conhecimento $K = (U, \mathbf{R})$ é chamada K-*conceitos*.

Seja $K = (U, \mathbf{R})$, entende-se por $IND(K)$ a família com todas as relações de equivalência definidas em K, ou seja, o conjunto mínimo contendo as relações de equivalência elementares adicionado pela relações de equivalência básicas possíveis.

Seja $K = (U, \mathbf{P})$ e $K' = (U, \mathbf{Q})$, K e K' são equivalentes se $IND(\mathbf{P}) = IND(\mathbf{Q})$, ou seja, se $U/\mathbf{P} = U/\mathbf{Q}$. Então, se $K \simeq K'$, K e K' contém os mesmos *conceitos* elementares. Caso $IND(\mathbf{P}) \subset IND(\mathbf{Q})$, diz-se que o conhecimento $\mathbf{P}$ é generalização de $\mathbf{Q}$ ou $\mathbf{Q}$ é especialização de $\mathbf{P}$.

Exemplo

Para uma melhor compreensão dos conceitos teóricos apresentados, considere o conjunto de objetos $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$ representando 10 clientes hipotéticos de uma empresa de energia elétrica. Estes objetos estão classificados da seguinte maneira:

- Tipo de Ligação
  - Monofásico: x_1, x_6, x_8.
  - Bifásico: x_3, x_4, x_7, x_{10}.
  - Trifásico: x_2, x_5, x_9.
- Classe
  - Residencial: x_1, x_4, x_7, x_{10}.
  - Rural: x_6, x_8.
  - Comercial: x_3, x_9.
  - Industrial: x_2, x_5.
- Curva de Consumo
  - Estável: $x_1, x_4, x_5, x_6, x_9, x_{10}$.
  - Instável: x_2, x_3, x_7, x_8.

Cada característica de objeto acima representa um conhecimento sobre U. Sendo assim, Tipo de Ligação, Classe e Curva de Consumo levam às relações de equivalência R_1, R_2 e R_3, respectivamente. Cada uma destas relações dá origem às seguintes classes de equivalência:

$$U/R_1 = \{\{x_1, x_6, x_8\}, \{x_3, x_4, x_7, x_{10}\}, \{x_2, x_5, x_9\}\}$$
$$U/R_2 = \{\{x_1, x_4, x_7, x_{10}\}, \{x_6, x_8\}, \{x_3, x_9\}, \{x_2, x_5\}\}$$
$$U/R_3 = \{\{x_1, x_4, x_5, x_6, x_9, x_{10}\}, \{x_2, x_3, x_7, x_8\}\}$$

Cada classe de equivalência ou subconjunto acima é um *conceito* elementar em $K = (U, \{R_1, R_2, R_3\})$. A intersecção entre os R_1-*conceitos* elementares e os R_3-*conceitos* elementares, por exemplo, leva aos $\{R_1, R_3\}$-*conceitos* básicos abaixo:

$$\{x_1, x_6, x_8\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_1, x_6\}$$
$$\{x_1, x_6, x_8\} \cap \{x_2, x_3, x_7, x_8\} = \{x_8\}$$

$$\{x_3, x_4, x_7, x_{10}\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_4, x_{10}\}$$
$$\{x_3, x_4, x_7, x_{10}\} \cap \{x_2, x_3, x_7, x_8\} = \{x_3, x_7\}$$

$$\{x_2, x_5, x_9\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_5, x_9\}$$
$$\{x_2, x_5, x_9\} \cap \{x_2, x_3, x_7, x_8\} = \{x_2\}$$

Sendo assim, $U/IND(\{R_1, R_3\}) = \{\{x_1, x_6\}, \{x_2\}, \{x_3, x_7\}, \{x_4, x_{10}\}, \{x_5, x_9\}, \{x_8\}\}$. A intersecção não-vazia entre todos *conceitos* elementares de R_1, R_2 e R_3 é:

$$\{x_1, x_6, x_8\} \cap \{x_1, x_4, x_7, x_{10}\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_1\}$$
$$\{x_1, x_6, x_8\} \cap \{x_6, x_8\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_6\}$$
$$\{x_1, x_6, x_8\} \cap \{x_6, x_8\} \cap \{x_2, x_3, x_7, x_8\} = \{x_8\}$$

$$\{x_3, x_4, x_7, x_{10}\} \cap \{x_1, x_4, x_7, x_{10}\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_4, x_{10}\}$$
$$\{x_3, x_4, x_7, x_{10}\} \cap \{x_1, x_4, x_7, x_{10}\} \cap \{x_2, x_3, x_7, x_8\} = \{x_7\}$$
$$\{x_3, x_4, x_7, x_{10}\} \cap \{x_3, x_9\} \cap \{x_2, x_3, x_7, x_8\} = \{x_3\}$$

$$\{x_2, x_5, x_9\} \cap \{x_3, x_9\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_9\}$$
$$\{x_2, x_5, x_9\} \cap \{x_2, x_5\} \cap \{x_1, x_4, x_5, x_6, x_9, x_{10}\} = \{x_5\}$$
$$\{x_2, x_5, x_9\} \cap \{x_2, x_5\} \cap \{x_2, x_3, x_7, x_8\} = \{x_2\}$$

Logo, $U/IND(\{R_1, R_2, R_3\}) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4, x_{10}\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}, \{x_9\}\}$. Comparando os *conceitos* de $U/IND(\{R_1, R_3\})$ e $U/IND(\{R_1, R_2, R_3\})$ fica claro que o aumento do conhecimento sobre U tende à aumentar o número de K-*conceitos*.

4.4 Rough Sets

Um *conceito* pode ser entendido como uma classificação, uma partição de objetos do universo. Porém, nem sempre, um *conceito* é *definível* para a base de conhecimento considerada.

Em outras palavras, muitas vezes não é possível definir uma classificação exata dos objetos à partir das relações de equivalência encontradas em uma base de conhecimento. Uma alternativa a este problema, o qual ficará mais evidente à seguir, é proposta por Rough Sets: encontrar *conceitos* (ou classificações) aproximados em uma base de conhecimento.

Seja $X \subseteq U$ um subconjunto representando uma classificação qualquer e R uma relação de equivalência, diz-se que X é R-*definível* se X é igual à união entre quaisquer R-*conceitos* básicos. Caso contrário, X é dito R-*indefinível*. Os conjuntos R-definíveis são também chamados R-*exatos* e os R-indefiníveis como R-*inexatos* ou R-*rough*.

Um subconjunto $X \subseteq U$ é dito *exato* em K se existir uma relação $R \in IND(K)$ tal que X seja R-exato. Da mesma maneira, X é dito *rough* se X é R-rough para todo $R \in IND(K)$.

4.4.1 Aproximações de Conjuntos

Dado $K = (U, \mathbf{R})$, um subconjunto $X \subseteq U$ e uma relação de equivalência $R \in IND(K)$, determina-se os seguintes subconjuntos:

$$\underline{R}X = \bigcup\{Y \in U/R : Y \subseteq X\}$$
$$\overline{R}X = \bigcup\{Y \in U/R : Y \cap X \neq \emptyset\}$$

em que $\underline{R}X$ é chamado de R-*aproximação inferior* de X e $\overline{R}X$ de R-*aproximação superior* de X. Ambas aproximações podem ser enunciadas da forma equivalente abaixo:

$$\underline{R}X = \{x \in U : [x]_R \subseteq X\}$$
$$\overline{R}X = \{x \in U : [x]_R \cap X \neq \emptyset\}$$

ou

$$x \in \underline{R}X \text{ se e somente se } [x]_R \subseteq X$$
$$x \in \overline{R}X \text{ se e somente se } [x]_R \cap X \neq \emptyset$$

O conjunto $\underline{R}X$ contem objetos de U que **com certeza** são classificados como objetos de X, considerando o conhecimento R. Ou seja, para um objeto de U pertencer ao conjunto $\underline{R}X$ o mesmo deve pertencer a um R-*conceito* que está contido em X.

Já o conjunto $\overline{R}X$ contem elementos que **possivelmente** são classificados como elementos de X, considerando o conhecimento R. Ou seja, para um objeto de U pertencer ao conjunto $\overline{R}X$ o mesmo deve estar em um R-*conceito* onde **pelo menos um** dos demais objetos deste *conceito* pertença a X.

Normalmente, outras notações que envolvem as aproximações são utilizadas, tais como:

- $POS_R(X) = \underline{R}X$ (ou R-*região positiva* de X): conjunto de objetos que com certeza são classificados como membros de X, considerando R.

- $NEG_R(X) = U - \overline{R}X$ (ou *R-região negativa* de X): conjunto de objetos que com certeza não são classificados como membros de X, considerando R.

- $BN_R(X) = \overline{R}X - \underline{R}X$ (ou *R-região de fronteira* de X): conjunto de objetos com indecisão quanto a sua classificação como membros de X e $-X$ (ou $U - X$), ou seja, somente com o conhecimento R não é possível afirmar que os mesmos são classificados em X ou $-X$. Esta indecisão acontece quando pelo menos um par de objetos pertencem à um mesmo *R-conceito*, porém somente um deles é elemento de X. Neste caso, ambos objetos pertencem à $BN_R(X)$ por indefinição exata de classificação.

Exemplo

Seja o conjunto U e as classes de equivalência de R_1, R_2 e R_3 apresentadas no exemplo da Seção 4.3:

$$U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$$

$$U/R_1 = \{\{x_1, x_6, x_8\}, \{x_3, x_4, x_7, x_{10}\}, \{x_2, x_5, x_9\}\}$$
$$U/R_2 = \{\{x_1, x_4, x_7, x_{10}\}, \{x_6, x_8\}, \{x_3, x_9\}, \{x_2, x_5\}\}$$
$$U/R_3 = \{\{x_1, x_4, x_5, x_6, x_9, x_{10}\}, \{x_2, x_3, x_7, x_8\}\}$$

Considerando o subconjunto X e a relação de equivalência $U/IND(\{R_1, R_3\})$, ora nomeada $R_{1,3}$:

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

$$R_{1,3} = \{\{x_1, x_6\}, \{x_2\}, \{x_3, x_7\}, \{x_4, x_{10}\}, \{x_5, x_9\}, \{x_8\}\}$$

Encontra-se as seguintes aproximações do subconjunto X:

$$\underline{R_{1,3}}X = \{x_1, x_2, x_6\}$$

$$\overline{R_{1,3}}X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_9, x_{10}\}$$

$$POS_{R_{1,3}}(X) = \{x_1, x_2, x_6\}$$

$$NEG_{R_{1,3}}(X) = \{x_8\}$$

$$BN_{R_{1,3}}(X) = \{x_3, x_4, x_5, x_7, x_9, x_{10}\}$$

Proposições

Da análise das aproximações de conjuntos acima, algumas proposições podem ser enunciadas:

1. X é R-definível se e somente se $\underline{R}X = \overline{R}X$.
2. X é rough com relação à R se e somente se $\underline{R}X \neq \overline{R}X$.
3. $\underline{R}X \subseteq X \subseteq \overline{R}X$
4. $X \subseteq Y$ implica que $\underline{R}X \subseteq \underline{R}Y$
5. $X \subseteq Y$ implica que $\overline{R}X \subseteq \overline{R}Y$
6. $\underline{R}(-X) = -\overline{R}X$
7. $\overline{R}(-X) = -\underline{R}X$

As provas das proposições acima, bem como de outras não enunciadas neste trabalho, encontram-se na referência base deste Capítulo (Pawlak, 1991).

4.4.2 Aproximações e Relações de Pertinência

Na teoria de conjuntos, a relação de pertinência entre elementos e conjuntos é sempre exata: dado um elemento a e um conjunto C, $a \in C$ ou $a \notin C$. Em outras palavras, cada elemento do universo é classificado como membro de X ou $-X$ necessariamente, para qualquer X. Portanto, considerando uma relação de pertinência, todo o conhecimento acerca dos objetos do universo é necessário para se obter *conceitos* exatos.

Com a aproximação de conjuntos proposta por Rough Sets, tem-se duas novas definições de relação de pertinência associada ao conhecimento disponível sobre o universo:

$$x \underline{\in}_R X \text{ se e somente se } x \in \underline{R}X$$
$$x \overline{\in}_R X \text{ se e somente se } x \in \overline{R}X$$

em que $\underline{\in}_R$ lê-se "x **com certeza pertence** a X com relação à R" e $\overline{\in}_R$ lê-se "x **possivelmente pertence** a X com relação à R". Estas novas relações são chamadas *relações de pertinência inferior e superior*, respectivamente.

Se $\underline{R}X = \overline{R}X$ (se X é R-exato), as relações de pertinência inferior e superior não são necessárias, visto que a relação de pertinência tradicional será suficiente para X.

4.4.3 Caracterização Numérica de Imprecisão

A região de fronteira $BN_R(X)$ evidencia a existência de imprecisão em X com relação à R. Quanto maior a cardinalidade do conjunto $BN_R(X)$, menor é a do conjunto $POS_R(X)$. Sendo assim, Rough Sets apresenta a seguinte *medida de precisão*:

$$\alpha_R(X) = \frac{|\underline{X}R|}{|\overline{X}R|}, X \neq \emptyset \tag{4.1}$$

Esta medida pode ser entendida como o grau de completude do conhecimento sobre X, com valor no intervalo $0 \leq \alpha_R(X) \leq 1$, para qualquer R. Se $\alpha_R(X) = 1$, então a região de fronteira de X com relação à R é vazia e X é R-definível. Se $\alpha_R(X) < 1$, então X é R-indefinível.

4.4.4 Caracterização Topológica de Imprecisão

Além da medida de precisão, que caracteriza numericamente o grau de imprecisão de um conjunto, Rough Sets apresenta também uma caracterização topológica de imprecisão. Dado um conjunto X, o mesmo pode ser classificado quanto às características das aproximações inferior e superior da seguinte forma:

1. Se $\underline{R}X \neq \emptyset$ e $\overline{R}X \neq U$, então X é chamado *rough R-definível*, pois com relação à R, existem elementos de U que com certeza são membros de X e $-X$;

2. Se $\underline{R}X = \emptyset$ e $\overline{R}X \neq U$, então X é chamado *internamente R-indefinível*, pois com relação à R, existem elementos de U que com certeza são membros de $-X$, porém não se tem certeza sobre elementos de U membros de X;

3. Se $\underline{R}X \neq \emptyset$ e $\overline{R}X = U$, então X é chamado *externamente R-indefinível*, pois com relação à R, existem elementos de U que com certeza são membros de X, porém não se tem certeza sobre elementos de U membros de $-X$;

4. Se $\underline{R}X = \emptyset$ e $\overline{R}X = U$, então X é chamado *totalmente R-indefinível*, pois como relação à R, não se pode ter certeza que elementos de U são membros de X ou $-X$.

4.5 Redução de Conhecimento

O conhecimento existente sobre um universo de objetos pode ser insuficiente ou mesmo excessivo. Quando insuficiente, leva a formação de *conceitos* R-indefiníveis e baixas medidas de precisão. Já quando é exagerado, é conveniente identificar aqueles conhecimentos que podem ser desconsiderados sem promover mudanças nos *conceitos*. Esta *redução de conhecimento* torna-se mais relevante quando o tamanho da base de conhecimento é limitado.

4.5.1 Reduto e Núcleo

Seja $\mathbf{R}$ uma família de relações de equivalência e $R \in \mathbf{R}$, R é dito *dispensável* em $\mathbf{R}$ se $IND(\mathbf{R}) = IND(\mathbf{R} - \{R\})$. Caso contrário, R é *indispensável* em $\mathbf{R}$. A família $\mathbf{R}$ é *independente* se cada $R \in \mathbf{R}$ é indispensável em $\mathbf{R}$. Caso contrário, $\mathbf{R}$ é *dependente*.

Seja $\mathbf{P} \subseteq \mathbf{R}$, o subconjunto $\mathbf{Q} \subseteq \mathbf{P}$ é *reduto* de $\mathbf{P}$ se $\mathbf{Q}$ é independente e $IND(\mathbf{Q}) = IND(\mathbf{P})$. Como podem existir relações dispensáveis e indispensáveis em $\mathbf{P}$, um reduto de $\mathbf{P}$ não é necessariamente único. O conjunto de todas relações indispensáveis em $\mathbf{P}$ é chamado *núcleo* de $\mathbf{P}$, ou simbolicamente $CORE(\mathbf{P})$. A relação entre reduto e núcleo é representada por:

$$CORE(\mathbf{P}) = \bigcap RED(\mathbf{P})$$

em que $RED(\mathbf{P})$ é a família de todos possíveis redutos de $\mathbf{P}$.

O núcleo pode ser visto com o conhecimento mais relevante acerca do universo, ou seja, compreende as relações que não podem ser eliminadas no processo de redução de conhecimento.

Exemplo

Seja o conjunto U e as classes de equivalência de $\mathbf{R} = \{P, Q, R\}$ e $IND(\mathbf{R})$ abaixo:

$$U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

$$U/P = \{\{x_1, x_2, x_7\}, \{x_3, x_8\}, \{x_4, x_5, x_6\}\}$$
$$U/Q = \{\{x_1\}, \{x_2, x_3, x_8\}, \{x_4, x_6, x_7\}\}$$
$$U/R = \{\{x_1, x_6, x_8\}, \{x_2, x_4\}, \{x_3, x_5, x_7\}\}$$

$$IND(\mathbf{R}) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}$$

Nenhuma relação de $\mathbf{R}$ individualmente é equivalente à $IND(\mathbf{R})$, ou seja, nenhuma relação de $\mathbf{R}$ é reduto de $\mathbf{R}$. A relação P é dispensável em $\mathbf{R}$ pois:

$$IND(\mathbf{R} - \{P\}) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\} = IND(\mathbf{R}).$$

A relação Q também é dispensável em $\mathbf{R}$ pois:

$$IND(\mathbf{R} - \{Q\}) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\} = IND(\mathbf{R}).$$

Já a relação R é indispensável em $\mathbf{R}$ pois:

$$IND(\mathbf{R} - \{R\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_8\}, \{x_4, x_6\}, \{x_5\}, \{x_7\}\} \neq IND(\mathbf{R}).$$

Logo, $RED(\mathbf{R}) = \{\{P, R\}, \{Q, R\}\}$ e $CORE(\mathbf{R}) = \{R\}$.

4.6 Representação do Conhecimento

As Seções anteriores abordaram exaustivamente o significado semântico do conhecimento como uma forma de classificação (ou partição) através de *conceitos*. Para uma melhor manipulação dos objetos e do conhecimento, utiliza-se um *Sistema de Representação do Conhecimento*, normalmente chamado de *Sistema de Informação (SI)*. Um SI é uma representação sintática do conhecimento sobre um conjunto de objetos e consiste de uma tabela de dados, onde as colunas são nomeadas como *atributos* e as linhas como *objetos*. Cada coluna representa uma relação de equivalência e cada linha armazena as classes de equivalência nas quais o objeto desta linha está inserido.

Exemplos

A Tabela 4.1 é um SI referente à base de conhecimento apresentada na Seção 4.3, em que as linhas representam clientes consumidores de energia elétrica e as colunas são alguns atributos destes clientes.

Tabela 4.1: SI de clientes consumidores de energia elétrica.

Clientes	Tipo de Ligação	Classe	Curva de Consumo
$c1$	Monofásico	Residencial	Estável
$c2$	Trifásico	Industrial	Instável
$c3$	Bifásico	Comercial	Instável
$c4$	Bifásico	Residencial	Estável
$c5$	Trifásico	Industrial	Estável
$c6$	Monofásico	Rural	Estável
$c7$	Bifásico	Residencial	Instável
$c8$	Monofásico	Rural	Instável
$c9$	Trifásico	Comercial	Estável
$c10$	Bifásico	Residencial	Estável

A Tabela 4.2 é um SI que ilustra a codificação das cores básicas do modelo de cores RGB. Neste modelo, combinando as cores vermelho (*red*), verde (*green*) e azul (*blue*) com intensidades entre 0 e 255, obtém-se outras cores derivadas destas primeiras.

Tabela 4.2: SI das cores básicas do modelo de cores RGB.

Cor	R (Red)	G (Green)	B (Blue)
Preto	0	0	0
Vermelho	255	0	0
Verde	0	255	0
Azul	0	0	255
Amarelo	255	255	0
Magenta	255	0	255
Ciano	0	255	255
Branco	255	255	255

4.6.1 Definição Formal

Formalmente, um SI é uma par $S = (U, A)$ em que:

U: conjunto finito e não-vazio chamado de *universo*.
A: conjunto finito e não-vazio de *atributos primitivos*.

em que todo atributo primitivo $a \in A$ é uma função $a : U \longrightarrow V_a$. O conjunto V_a é chamado *conjunto de valores* ou *domínio* de a.

Dado qualquer subconjunto $B \subseteq A$, define-se uma *relação de indiscernibilidade* como sendo:

$$IND(B) = \{(x, y) \in U^2 : \text{para todo } a \in B, a(x) = a(y)\}$$

ou

$$IND(B) = \bigcap_{a \in B} IND(a)$$

Um subconjunto $B \subseteq A$ também é chamado de atributo. Caso B seja um conjunto com um único elemento, B é chamado *atributo primitivo*. Caso contrário, é chamado de *atributo composto*.

Todas as definições apresentadas para uma base de conhecimento $K = (U, \mathbf{R})$ podem ser expressadas em definições para um SI $S = (U, A)$ da seguinte forma: se $R \in \mathbf{R}$ e $U/R = \{X_1, ..., X_k\}$, então no conjunto de atributos A estão contidos os atributos $a_R : U \longrightarrow V_{a_R}$, tal que $V_{a_R} = \{1, ..., k\}$ e $a_R(x) = i$ se e somente se $x \in X_i$ para $i = 1, ..., k$.

Para ilustrar este mapeamento de base de conhecimento para SI, considere a Tabela 4.3.

Cada linha representa um objeto do universo $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ e cada coluna um atributo de $A = \{a, b, c, d\}$, onde $V_a = V_b = V_c = V_d = \{1, 2, 3\}$. A partir de subconjuntos de A, encontra-se partições sobre os objetos de U, tais como:

Tabela 4.3: SI qualquer.

U	a	b	c	d
1	1	2	1	2
2	3	3	3	1
3	2	1	2	3
4	2	3	3	3
5	3	1	3	1
6	2	2	1	2
7	1	1	2	2
8	3	3	2	1
9	1	2	1	2
10	2	2	3	3

$$U/IND\{a\} = \{\{1,7,9\},\{2,5,8\},\{3,4,6,10\}\}$$
$$U/IND\{b\} = \{\{1,6,9,10\},\{2,4,8\},\{3,5,7\}\}$$
$$U/IND\{c\} = \{\{1,6,9\},\{2,4,5,10\},\{3,7,8\}\}$$

$$U/IND\{a,b\} = \{\{1,9\},\{2,8\},\{3\},\{4\},\{5\},\{6,10\},\{7\}\}$$
$$U/IND\{a,c\} = \{\{1,9\},\{2,5\},\{3\},\{4,10\},\{6\},\{7\},\{8\}\}$$
$$U/IND\{b,c\} = \{\{1,6,9\},\{2,4\},\{3,7\},\{5\},\{8\},\{10\}\}$$

$$U/IND\{a,b,c\} = \{\{1,9\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{10\}\}$$
$$U/IND(A) = \{\{1,9\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{10\}\}$$

Como as partições determinadas por $U/IND(A)$ e $U/IND\{a,b,c\}$ são idênticas, o atributo d é dito dispensável. Da mesma forma, como as partições de $U/IND\{a,b\}$, $U/IND\{a,c\}$ e $U/IND\{b,c\}$ são distintas de $U/IND\{a,b,c\}$, os atributos a, b e c são indispensáveis e compõem o reduto $P = \{a,b,c\}$.

Considerando o subconjunto de atributos $Q = \{b,c\}$ e o subconjunto de objetos $X = \{1,2,3,4,5\}$, define-se as seguintes aproximações de X:

$$\underline{Q}X = \{2,4,5\}$$
$$\overline{Q}X = \{1,2,3,4,5,6,7,9\}$$
$$POS_Q(X) = \{2,4,5\}$$
$$NEG_Q(X) = \{8,10\}$$
$$BN_Q(X) = \{1,3,6,7,9\}$$

4.6.2 Matriz de Discernimento

A *matriz de discernimento* é uma tabela construída com o objetivo de encontrar os redutos e o core de um SI. Seja $S = (U, A)$, onde $U = \{x_1, x_2, ..., x_n\}$, a matriz de discernimento $M(S)$ é uma tabela $n \times n$ em que cada célula é preenchida da seguinte forma:

$$(c_{ij}) = \{a \in A : a(x_i) \neq a(x_j)\} \text{ para } i, j = 1, 2, ..., n.$$

Em outras palavras, cada objeto x_i de U é comparado com os demais objetos x_j. Aqueles atributos que possuírem valores distintos em x_i e x_j farão parte de c_{ij}.

O reduto $B \subseteq A$ é o subconjunto mínimo de A tal que:

$$B \cap c \neq \emptyset \text{ para qualquer } c \neq \emptyset \text{ em } M(S).$$

Caso uma célula c_{ij} seja preenchida com um único atributo a, este atributo fará parte de $CORE(A)$, ou seja:

$$CORE(A) = \{a \in A : c_{ij} = (a)\}, \text{ para algum } i, j.$$

Exemplo

A Tabela 4.4 é a matriz de discernimento do SI da Tabela 4.3. Nota-se claramente que o objeto 9 foi excluído da tabela (não há linha nem coluna 9). Isto porquê os objetos 1 e 9 são semelhantes, não necessitando que ambos sejam comparados aos demais objetos. Observa-se também que a matriz de discernimento será sempre simétrica, sendo suficiente preencher c_{ij} ou c_{ji}. As células c_{ii}, naturalmente, serão sempre vazias.

A partir da matriz de discernimento, identifica-se que $CORE(A) = RED(A) = \{a, b, c\}$.

Tabela 4.4: Matriz de discernimento do SI da Tabela 4.3.

$c_{i,j}$	1	2	3	4	5	6	7	8	10
1									
2	a, b, c, d								
3	a, b, c, d	a, b, c, d							
4	a, b, c, d	a, d	b, c						
5	a, b, c, d	b	a, c, d	a, b, d					
6	a	a, b, c, d	b, c, d	b, c, d	a, b, c, d				
7	b, c	a, b, c, d	a, d	a, b, c, d	a, c, d	a, b, c			
8	a, b, c, d	c	a, b, d	a, c, d	b, c	a, b, c, d	a, b, d		
10	a, c, d	a, b, d	b, c	b	a, b, d	c, d	a, b, c, d	a, b, c, d	

4.7 Tabelas de Decisão

Um SI normalmente é acrescido de pelo menos um atributo, o qual realiza uma classificação sobre os objetos, levando à tomada de decisões. Os SI incrementados por atributos de decisão são chamados *Tabelas de Decisão*. Tais tabelas permitem que objetos dêem origem à regras de decisão, possibilitando a aplicação do conhecimento dos objetos existentes na classificação de novos objetos.

4.7.1 Definição Formal

Seja $C, D \subset A$ dois subconjuntos chamados *atributos condicionais* e *atributos de decisão*, respectivamente, define-se por Tabela de Decisão o SI da forma $T = (U, A, C, D)$.

Para cada objeto $x \in U$ associa-se uma função $d_x : A \longrightarrow V$, chamada *regra de decisão* em T, tal que $d_x(a) = a(x)$, para todo $a \in C \cup D$.

Uma regra de decisão d_x possui duas partes básicas: as *condições* e as *decisões*. As condições são restrições derivadas de atributos condicionais e são denotadas por $d_x|C$. Da mesma forma, as decisões são restrições derivadas de atributos de decisão e são denotadas por $d_x|D$. Uma regra d_x é *consistente* em T se, para todo $x \neq y$, $d_x|C = d_x|C$ implica que $d_x|D = d_x|D$. Caso contrário, a regra d_x é dita *inconsistente*.

Exemplo

A Tabela de Decisão abaixo ilustra o SI da Tabela 4.3, incrementado de e. Sendo assim os atributos a, b, c e d são chamados atributos condicionais, enquanto que e é chamado atributo de decisão. Considerando a Tabela 4.5, as regras de decisão 1 e 9 são inconsistentes, enquanto que as demais são consistentes.

Tabela 4.5: Tabela de Decisão da Tabela 4.3.

U	a	b	c	d	e
1	1	2	1	2	1
2	3	3	3	1	3
3	2	1	2	3	4
4	2	3	3	3	4
5	3	1	3	1	3
6	2	2	1	2	1
7	1	1	2	2	2
8	3	3	2	1	3
9	1	2	1	2	2
10	2	2	3	3	4

4.7.2 Redução em Tabelas de Decisão

Tabelas de Decisão são utilizadas em várias aplicações, envolvendo problemas de classificação, reconhecimento de padrão, sistemas especialistas, etc. Normalmente, estas tabelas são submetidas a processos de redução ou simplificação, dentre eles:

1. Redução de atributos condicionais: obtida através do cômputo do reduto, permitindo que atributos dispensáveis sejam removidos;

2. Eliminação de regras duplicadas: após selecionar os atributos condicionais de um reduto, linhas ou regras de decisão podem tornar-se idênticas, sendo suficiente manter apenas uma regra representante;

3. Redução de valores de atributos condicionais: é possível que uma regra seja simplificada através da eliminação de restrições condicionais, visto que eventualmente nem todas condições de uma regras necessitam ser testadas para realizar-se uma decisão.

Para realizar-se uma redução em Tabelas de Decisão por eliminação de valores de atributos condicionais, utiliza-se um método semelhante àquele empregado na identificação de redutos em SI.

Seja $B \subseteq A$ um subconjunto de atributos e x um objeto qualquer, $[x]_B$ consiste da classe de equivalência determinada por $IND(B)$ que contém o objeto x. Então, a partir de um subconjunto de atributos condicionais C de uma regra d_x, encontra-se $[x]_C = \bigcap_{a \in C}[x]_a$. Eliminar valores de atributos condicionais consiste em eliminar classes de equivalência supérfluas $[x]_a$ da classe de equivalência $[x]_C$.

O exemplo abaixo ilustra as etapas do método tradicional de redução em Tabelas de Decisão.

Exemplo

Considerando a Tabela 4.5 como sendo a Tabela de Decisão a ser reduzida, o primeiro passo empregado é a eliminação de atributos supérfluos que não pertencem ao reduto. Como este reduto já foi computado pela matriz de discernimento da Tabela 4.4, o mesmo foi empregado dando origem à Tabela 4.6.

Pela ausência de linhas ou regras duplicadas na Tabela 4.6, procede-se com a redução de valores de atributos condicionais.

O procedimento consiste em, para cada regra de decisão, encontrar o core e os redutos que permitem a regra manter a mesma decisão sobre os objetos. Como as regras 1 e 9 são inconsistentes, as mesmas não sofrem redução. Tomando como exemplo as regras 2 e 3 tem-se, respectivamente:

Tabela 4.6: Eliminação do reduto da Tabela 4.5.

U	a	b	c	e
1	1	2	1	1
2	3	3	3	3
3	2	1	2	4
4	2	3	3	4
5	3	1	3	3
6	2	2	1	1
7	1	1	2	2
8	3	3	2	3
9	1	2	1	2
10	2	2	3	4

$$[2]_{\{a,b,c\}} = [3]_a \cap [3]_b \cap [3]_c = \{2,5,8\} \cap \{2,4,8\} \cap \{2,4,5,10\} = \{2\}$$
$$[2]_e = \{2,5,8\}$$

$$[3]_{\{a,b,c\}} = [2]_a \cap [1]_b \cap [2]_c = \{3,4,6,10\} \cap \{3,5,7\} \cap \{3,7,8\} = \{3\}$$
$$[3]_e = \{3,4,10\}$$

A igualdade $[2]_{\{a,b,c\}} = \{2\}$ indica que a regra 2 é consistente, enquanto $[2]_e = \{2,5,8\}$ ilustra as regras que possuem a mesma decisão da regra 2. A mesma consideração pode ser feita para a regra 3.

Para encontrar o core da regra 2, por exemplo, analisa-se a eliminação de cada condição da regra:

$$[3]_a \cap [3]_b = \{2,8\}$$
$$[3]_a \cap [3]_c = \{2,5\}$$
$$[3]_b \cap [3]_c = \{2,4\}$$

O atributo a é core da regra 2 pois $[3]_b \cap [3]_c = \{2,4\} \nsubseteq \{2,5,8\}$. Procedendo da mesma forma, encontra-se os cores das demais regras consistentes, os quais são mostrados na Tabela 4.7.

As regras foram reagrupadas na Tabela 4.7 de acordo com suas classes de decisão e separadas em consistentes e inconsistentes. A partir dos cores encontrados, identifica-se os possíveis redutos de cada linha, como exemplificado abaixo para as regras 2 e 3, respectivamente:

Tabela 4.7: Core das regras da Tabela 4.6.

U	a	b	c	e
2	3	-	-	3
5	-	-	-	3
8	-	-	-	3
3	2	-	-	4
4	2	-	-	4
10	-	-	3	4
6	2	-	1	1
7	1	-	-	2
1	1	2	1	1
9	1	2	1	2

$$[3]_a = [2]_e = \{2, 8\}$$

$$[2]_a \cap [1]_b = \{3, 4, 6, 10\} \cap \{3, 5, 7\} = \{3\} \subseteq \{3, 4, 10\}$$
$$[2]_a \cap [2]_c = \{3, 4, 6, 10\} \cap \{3, 7, 8\} = \{3\} \subseteq \{3, 4, 10\}$$

Após aplicar as etapas acima para as demais regras, deriva-se uma nova regra de cada possível reduto, alcançando a Tabela 4.8.

Tabela 4.8: Core das regras da Tabela 4.6.

U	a	b	c	e
2	3	-	-	3
5	3	-	-	3
8	3	-	-	3
3	2	1	-	4
3’	2	-	2	4
4	2	3	-	4
4’	2	-	3	4
10	2	-	3	4
10’	-	2	3	4
6	2	-	1	1
7	1	1	-	2
7’	1	-	2	2
1	1	2	1	1
9	1	2	1	2

Nota-se na Tabela 4.8 que as regras que possuíam mais de um reduto, deram origem à novas regras, sendo suficiente considerar qualquer uma delas como representante da regra original.

Para minimizar a Tabela de Decisão reduzida, é conveniente tomar as regras duplicadas, ou seja, aquelas que possuem as mesmas condições e levam à mesma decisão. Isto porque, na existência de regras duplicadas, somente uma delas é suficiente para a Tabela de Decisão. Seguindo esta consideração, chega-se à Tabela 4.9, a qual representa a redução da Tabela de Decisão ilustrada pela Tabela 4.5.

Tabela 4.9: Redução final da Tabela 4.5.

U	a	b	c	e
2,5,8	3	-	-	3
3	2	1	-	4
4',10	2	-	3	4
6	2	-	1	1
7	1	1	-	2
1	1	2	1	1
9	1	2	1	2

Para fins de tomada de decisão, não é necessário manter a numeração original das regras. Sendo assim, a Tabela 4.9 pode ser reordenada como na Tabela 4.10.

Tabela 4.10: Reordenação da Tabela 4.9.

U	a	b	c	e
1	2	-	1	1
2	1	2	1	1
3	1	1	-	2
4	1	2	1	2
5	3	-	-	3
6	2	1	-	4
7	2	-	3	4

4.8 Considerações Finais

Neste capítulo foi apresentada a abordagem teórica ou matemática de Rough Sets, baseada em (Pawlak, 1991). Os conceitos formais apresentados ajudam a compreender Rough Sets em sua essência, fortalecendo a abordagem prática do Capítulo 3.

No próximo capítulo é realizado o pré-tratamento e a consolidação dos dados, os quais serão submetidos posteriormente aos conceitos de Rough Sets na mineração de padrões de comportamento fraudulentos.

CAPÍTULO

5

Pré-Tratamento e Consolidação dos Dados

5.1 Introdução

O pré-tratamento, como dito na Subseção 2.4.3, normalmente é a etapa mais demorada em um processo de descoberta de conhecimento em banco de dados (Manilla, 1994). Esta afirmação confirmou-se neste trabalho.

O banco de dados de uma empresa de distribuição de energia elétrica contém inúmeras informações, desde o histórico de consumo dos clientes à dados técnicos dos dispositivos de distribuição. Enfim, uma grande massa de dados que requer segurança e confiabilidade, tanto no acesso como no armazenamento e na recuperação de informações.

A tarefa de selecionar quais tabelas, registros e atributos do banco de dados serão estudados é fundamental no processo de descoberta de conhecimento. Principalmente porque, nas etapas iniciais, não se sabe exatamente quais informações são excessivas e quais são imprescindíveis.

Inicialmente, é apresentado na Seção 5.2 um descritivo das tabelas que compõem o banco de dados utilizado, enunciando cada atributo disponível. Posteriormente, na Seção 5.3, são apresentadas as etapas de pré-tratamento utilizadas na consolidação dos dados para mineração. Ao final, na Seção 5.4, são feitas as considerações finais do capítulo.

5.2 Descrição do Banco de Dados

Para este trabalho, foi acessado parte de um banco de dados, do período de novembro de 2002 à outubro de 2003. Esta amostra consiste de um arquivo do *Microsoft Access*[1] que contém três tabelas, cujos atributos estão enunciados abaixo:

1. **Tabela** $Consumo$

   - $Cons\_Id$: identificação única para cada unidade consumidora (ou cliente), também chamado de CDC. É um atributo do tipo cadeia de caracteres (por exemplo "01.010.10.101010");
   - $Cons\_Mes$: ano e mês das informações contidas no registro. Consiste de um tipo numérico com seis algarismos, identificando nos quatro primeiros o ano e nos dois últimos o mês (por exemplo 200212 e 200305);
   - $Cons\_Munic$: identificador numérico que representa o município onde a unidade consumidora está localizada (por exemplo 75);
   - $Cons\_Ativ$: código numérico que enquadra a unidade consumidora em alguma atividade, tendo maior distinção entre clientes comerciais e industriais (por exemplo 1109);
   - $Cons\_Tarifa$: informação da classe (residencial, comercial, industrial, etc) e do tipo de ligação (monofásica, bifásica, trifásica ou primária) concatenadas em um único identificador do tipo cadeia de caracteres (por exemplo "02.10.22");
   - $Cons\_Trafo$: identificação numérica do trafo (ou poste) ao qual a unidade consumidora está conectada (por exemplo 123456789012);
   - $Cons\_Cons$: quantidade de energia elétrica consumida em KWh, no mês e ano de referência do registro (por exemplo 280).

2. **Tabela** $Inspecao$

   - $Insp\_Id$: utilizado para relacionar um registro de $Inspecao$ a uma unidade consumidora de $Consumo$. Desta forma, armazena a mesma informação do atributo $Cons\_Id$;
   - $Insp\_Data$: atributo que armazena o dia, o mês e o ano em que ocorreu uma inspeção, no formato data (por exemplo 09/25/2003);
   - $Insp\_Result$: cadeia de caracteres enunciando o resultado da inspeção (por exemplo "FRAUDE").

[1] http://office.microsoft.com/access

3. **Tabela** $Trafos$

   - $Traf\_Trafo$: identificação única para cada trafo, permitindo um relacionamento com o atributo $Cons\_Trafo$ da tabela $Consumo$. Também está armazenado como um atributo numérico (por exemplo 749052726984);
   - $Traf\_Mes$: ano e mês das informações contidas no registro, sendo semelhante ao atributo $Cons\_Mes$.
   - $Traf\_Cons$: quantidade de energia elétrica consumida em KWh pelas unidades consumidoras conectadas no trafo, no mês e ano de referência do registro (por exemplo 10610).

5.3 Etapas de Pré-Tratamento dos Dados

A tabela $Consumo$ trás como principal informação o consumo de energia elétrica de cada cliente, mês a mês, no período de novembro de 2002 à outubro de 2003. Esperava-se, portanto, que cada cliente tivesse 12 registros, um para cada mês do período amostrado. Porém, a tabela $Consumo$ possui 7,266.819 registros e 642.720 clientes distintos, uma média de 11,3 registros por cliente. Notou-se então que em $Consumo$ existem clientes com menos de 12 registros e outros com mais de 12 registros.

Já a tabela $Inspecao$ possui 81.942 registros, cada um representando uma inspeção realizada em um conjunto de 64.326 clientes distintos. Do total de clientes inspecionados, 49.514 sofreram uma única inspeção e 14.812 sofreram pelo menos duas, entre novembro de 2002 à outubro de 2003.

A tabela $Trafos$ possui 326.748 registros, cada um representando o consumo de energia elétrica em um dado trafo, no mês em questão. De um total de 42.040 trafos distintos, 29.286 possuem menos ou mais de 12 registros.

As informações quantitativas para $Consumo$, $Inspecao$ e $Trafos$ estão simplificadas na Tabela 5.1.

Tabela 5.1: Informações quantitativas do banco de dados.

Tabela	Número de registros	Elementos distintos
$Consumo$	7,266.819	642.720
$Inspecao$	81.942	64.326
$Trafos$	326.748	42.040

Após a avaliação inicial do banco de dados, inicia-se um conjunto de etapas de pré-tratamento dos dados, nas quais foram realizadas operações específicas sobre as tabelas $Consumo$, $Inspecao$

e $Trafo$. As subseções seguintes apresentam o descritivo de cada uma destas etapas, mostrando quais decisões foram tomadas na permanência e na eliminação de dados.

5.3.1 Relacionamento de Consumo com Inspeção

A primeira tarefa efetuada foi o relacionamento entre registros das tabelas $Consumo$ e $Inspecao$, ou seja, verificar se há inspeção para um determinado cliente em algum mês do período de amostragem. Quando uma inspeção foi relacionada à um cliente, o resultado da mesma foi adicionado à tabela $Consumo$, derivando uma nova tabela chamada CI. Portanto, a tabela CI contem todos os registros (mês a mês) das unidades consumidoras que receberam pelo menos uma inspeção, com o acréscimo do resultado desta inspeção no registro em que o mês de consumo coincide com a data da inspeção. Um resultado de inspeção nulo foi inserido nos registros com meses em que o cliente não recebeu inspeção. A tabela CI, além de receber os resultados de inspeção da tabela $Inspecao$, manteve todos os demais atributos contidos em $Consumo$. Por este motivo, tanto $Consumo$ quanto $Inspecao$ deixaram de ser necessárias nas etapas seguintes, sendo substituídas apenas por CI.

Outra tarefa executada nesta etapa foi a decodificação do atributo CI_Tarifa em CI_Cls e CI_TLig. O novo atributo CI_Cls corresponde aos dois primeiros algarismos de CI_Tarifa e informa a que classe de serviço o cliente pertence, dentre elas: residencial (1), comercial (2), industrial (3), poder público (4), etc. Já o atributo CI_TLig corresponde ao dois algarismos finais de CI_Tarifa e informa qual o tipo de ligação do cliente, ou seja: primária, monofásica, bifásica e trifásica. Extraído estes atributos de CI_Tarifa, o mesmo também deixou de ser necessário nas etapas seguintes.

O atributo CI_DCons foi criado à partir de CI_Cons, representando a variação de energia elétrica consumida pelo cliente, ou seja, o consumo no mês do registro menos o consumo no mês anterior. Obviamente, valores negativos de CI_DCons indicam que o cliente diminuiu o consumo em relação ao mês anterior. Um valor nulo foi inserido no primeiro registro, pois o mesmo não possui registro anterior para a subtração.

Objetivando um melhor entendimento das operações realizadas nesta etapa, a Tabela 5.2 ilustra alguns atributos da tabela CI, tomando como exemplo os registros de uma unidade consumidora anônima.

Ao final desta etapa, a tabela CI possuía 659.462 registros, distribuídos por 59.489 unidades consumidoras distintas. Sendo assim, dos 64.326 clientes distintos da tabela $Inspecao$, 4.837 (7,5%) não se relacionaram com clientes da tabela $Consumo$. Esta diferença ocorreu por dois motivos:

1. um cliente da tabela $Inspecao$ não está registrado como cliente da tabela $Consumo$;

Tabela 5.2: Registros de uma unidade consumidora anônima da tabela CI.

CI_Id	CI_Mes	CI_TLig	CI_Cls	CI_Cons	CI_DCons	CI_Result
0.000.00.00000	200211	23	2	570		NORMAL
0.000.00.00000	200212	23	2	700	130	
0.000.00.00000	200301	23	2	590	-110	
0.000.00.00000	200302	23	2	640	50	
0.000.00.00000	200303	23	2	550	-90	
0.000.00.00000	200304	23	2	630	80	
0.000.00.00000	200305	23	2	510	-120	
0.000.00.00000	200306	23	2	480	-30	
0.000.00.00000	200307	23	2	460	-20	
0.000.00.00000	200308	23	2	660	200	
0.000.00.00000	200309	23	2	470	-190	NORMAL
0.000.00.00000	200310	23	2	540	70	

2. há registros para um dado cliente em ambas as tabelas, porém o mês da inspeção não coincide com o mês registrado em $Consumo$. Sendo assim o cliente não terá nenhuma inspeção e nenhum de seus registros na tabela CI.

5.3.2 Relacionamento de Consumo e Inspeção com Trafos

Para relacionar as tabelas CI e $Trafos$, adicionando a cada registro de cliente o consumo do trafo em que o mesmo está conectado, utilizam-se as informações dos códigos dos trafos e do mês de referência. Ao avaliar o atributo CI_Trafo, foram encontrados 33.771 registros com valor “NAO SE APLICA”, os quais foram descartados pela impossibilidade de relacionamento com $Trafos$. A tabela CI passou a ter 625.691 registros e 57.334 unidades consumidoras distintas.

O relacionamento entre CI e $Trafos$, chamado CIT, possui 473.152 registros e 47.987 unidades consumidoras. Esta redução considerável do número de registros de CIT em relação à CI (24%) ocorreu por fatores semelhantes aos da Subseção 5.3.1:

1. um código de trafo (CI_Trafo) ou mês de referência (CI_Mes) em CI não possui intersecção em $Trafos$;

2. um registro de CI com resultado de inspeção não-nulo não possui associação com nenhum registro de $Trafos$, levando a eliminação dos demais registros da unidade consumidora com resultado de inspeção nulo.

5.3.3 Concentração de Registros

A tabela CIT compreende atributos originais e derivados de $Consumo$, $Inspecao$ e $Trafos$, além de um conjunto de registros para cada unidade consumidora. Com o intuito de manter apenas um registro para cada unidade consumidora, primeiramente os clientes foram agrupados pela quantidade de meses (ou registros) que possuem em CIT. O resultado deste agrupamento pode ser visto na Tabela 5.3. A maioria das unidades consumidoras (67%) possuem 10 registros, que é praticamente a média de registros por clientes distintos em CIT (9,85). Em contrapartida, há apenas 3 unidades consumidoras acima de 16 registros.

Tabela 5.3: Unidades consumidoras da tabela CIT agrupadas pelo número de registros.

Número de registros ou meses	Número de unidades consumidoras
01	280
02	119
03	231
04	391
05	616
06	753
07	989
08	1.152
09	1.568
10	32.329
11	5.813
12	3.521
13	157
14	40
15	15
16	10
18	1
19	1
20	1

Após o agrupamento, foram descartados os clientes com número de registros menor que 4 e maior que 16, eliminando de CIT 1.268 registros de 633 clientes distintos. Também foram removidos 7.263 registros de 807 clientes, os quais possuíam pelo menos um mês com valores negativos para o atributo (CIT_Cons), sendo que o consumo mínimo esperado é zero. A tabela CIT, após as eliminações acima, passou a ter 464.621 registros de 46.547 clientes distintos.

A concentração das informações de clientes em um único registro é feita tomando qualquer um dos valores dos atributos estáticos e realizando alguma operação sobre os atributos dinâmicos, pois os mesmos variam seus valores, mês a mês, para cada cliente. Os atributos dinâmicos de CIT são:

1. consumo de energia elétrica do cliente no mês (CIT_Cons);
2. variação de consumo de energia elétrica em relação ao mês anterior (CIT_DCons);
3. consumo de energia elétrica do trafo no mês (CIT_TCons).

Os três atributos dinâmicos deram origem aos cinco novos atributos abaixo, os quais possuem um único valor para cada unidade consumidora:

1. CIT_Cmedia: média entre os valores de (CIT_Cons), representando a média de consumo do cliente;
2. CIT_Cdp: desvio-padrão entre os valores de (CIT_Cons), representando o desvio-padrão do consumo do cliente;
3. CIT_Tmedia: média entre os valores de (CIT_TCons), representando a média de consumo do trafo em que o cliente está conectado;
4. CIT_Tdp: desvio-padrão entre os valores de (CIT_TCons), representando o desvio-padrão do consumo do trafo;
5. CIT_Delta_Cmax: valor mínimo de (CIT_DCons), representando a diminuição máxima do consumo de energia.

5.3.4 Seleção de Clientes Normais e Fraudadores

A tabela CIT passou a concentrar em 46.547 registros, um para cada cliente, todas as informações desejadas sobre as unidades consumidoras. A Tabela 5.4 ilustra a quantidade de clientes para cada possível resultado de inspeção. Como o objetivo deste trabalho é detectar os clientes fraudulentos, somente aqueles que possuem resultado "NORMAL" ou "FRAUDE" foram selecionados. Desta forma, a tabela CIT foi renomeada para CIT_NF e passou a ter 41.290 registros, sendo 95,4% de clientes normais e 4,6% de fraudadores.

A tabela CIT_NF faz parte de um banco de dados do *Microsoft Access*, juntamente com as demais tabelas intermediárias ao pré-tratamento de dados. Porém, o processo de descoberta de conhecimento foi realizado usando-se o programa *MATLAB*[2], o qual apresenta várias ferramentas para a manipulação de matrizes (que podem ser vistas como tabelas). Sendo assim, os atributos da tabela CIT_NF foram importados para o *MATLAB*, onde cada atributo é um vetor numérico ou de caracteres com 41.290 elementos.

Uma última eliminação de clientes foi realizada sobre os registros (ou linhas no *MATLAB*) que apresentaram valor zero para média de consumo do cliente ou do trafo. Esta remoção não foi realizada na tabela CIT_NF do *Microsoft Access* pois acreditava-se que registros com médias

[2]http://www.mathworks.com/

Tabela 5.4: Unidades consumidoras da tabela CIT agrupadas pelos resultados de inspeção.

Resultado de Inspeção	Número de unidades consumidoras
NORMAL	39.389
FRAUDE	1.901
FALHA DE MEDICAO	1.821
IRREGULARIDADE COMERCIAL	1.518
IMPEDIMENTO	1.432
AUTORELIGAMENTO	426
IRREGULARIDADE TECNICA	60

nulas seriam importantes no processo de mineração, o que não foi comprovado posteriormente. O tamanho final dos vetores de atributos no *MATLAB* é de 40.492 elementos, onde 38.621 (95,4%) possuem resultado de inspeção normal, enquanto 1.871 (4,6%) apresentam resultado fraudulento.

A tabela 5.5 apresenta o conjunto de atributos disponíveis para o processo de mineração, informando seus possíveis valores e a que tipo ou classe do *MATLAB* pertencem.

Tabela 5.5: Conjunto de atributos disponíveis para o processo de mineração.

Nº	Atributos	Valores Distintos	Tipo	Distribuição
1	Id	40.492	Texto	Categórico
2	$Resultado\_Str$	NORMAL ou FRAUDE	Texto	Categórico
3	$Resultado\_Num$	2	Numérico	Categórico
4	$Atividade$	449	Numérico	Categórico
5	$Classe$	8	Numérico	Categórico
6	$Tipo\_Lig$	4	Numérico	Categórico
7	$Municipio$	72	Numérico	Categórico
8	$Media\_Consumo$	12.834	Numérico	Contínuo
9	$Dp\_Consumo$	35.171	Numérico	Contínuo
10	$Delta\_Consumo$	32.250	Numérico	Contínuo
11	$Media\_Trafo$	14.242	Numérico	Contínuo
12	Dp_Trafo	14.253	Numérico	Contínuo

5.4 Considerações Finais

Neste capítulo foi realizado o pré-tratamento e a consolidação dos dados para a aplicação da metodologia de detecção de fraudes usando Rough Sets. Primeiramente foi feita uma descrição do banco de dados utilizado, enunciando as tabelas e atributos disponíveis. Posteriormente,

realizou-se as etapas de pré-tratamento dos dados, os quais foram consolidados e disponibilizados para mineração.

No próximo capítulo é apresentada a metodologia propostas para detecção de fraudes em consumidores de energia elétrica.

CAPÍTULO

6

Metodologia para Detecção de Fraudes Usando Rough Sets

6.1 Introdução

O pré-tratamento do banco de dados teve como resultado 12 vetores de atributos, os quais foram apresentados na Tabela 5.5 do Capítulo 5. A partir destes atributos e dos conceitos de Rough Sets explorados no decorrer deste trabalho, desenvolveu-se uma metodologia para detecção de fraudes em unidades consumidoras de energia elétrica.

A metodologia é apresentada na Seção 6.2, seguida por seu teste de confiabilidade na Seção 6.3. Certificada a eficiência da metodologia, alguns conjuntos de atributos são avaliados na Seção 6.4, na busca pelas melhores características que definem os clientes fraudadores, de acordo com as medidas de avaliação empregadas. Ao final, na Seção 6.5, são feitas as considerações finais do capítulo.

6.2 Metodologia

Como no pré-tratamento dos dados no Capítulo 5, a metodologia para detecção de fraudes é dividida em etapas, as quais são enunciadas nas subseções seguintes. Estas etapas englobam tanto conceitos de Rough Sets como procedimentos típicos de mineração de dados.

6.2.1 Discretização de Atributos

O primeiro passo antes da aplicação dos conceitos de Rough Sets seria a reunião de um conjunto de atributos em uma única tabela, chamada Tabela de Decisão. Porém, cada atributo desta tabela deve ser categórico, ou seja, ter um conjunto finito de valores. Os atributos contínuos disponíveis estão ilustrados novamente na Tabela 6.1. Embora todos eles possuam uma quantidade de valores distintos menor que o próprio tamanho do vetor (40.492 elementos), estes atributos estão no domínio dos números reais, podendo admitir infinitos valores. Caso estes atributos contínuos fossem inseridos em uma Tabela de Decisão sem serem discretizados, valores praticamente iguais seriam tratados como distintos. Por exemplo, os valores 350,0 e 350,5 para o atributo $Media\_Consumo$, apesar de representarem a mesma informação para a média de consumo, seriam tratados como valores distintos nas comparações dos algoritmos de Rough Sets.

Tabela 6.1: Atributos contínuos a serem discretizados.

Nº	Atributos	Valores Distintos	Tipo	Distribuição
1	$Media\_Consumo$	12.834	Numérico	Contínuo
2	$Dp\_Consumo$	35.171	Numérico	Contínuo
3	$Delta\_Consumo$	32.250	Numérico	Contínuo
4	$Media\_Trafo$	14.242	Numérico	Contínuo
5	Dp_Trafo	14.253	Numérico	Contínuo

Para discretizar estes atributos, implementou-se uma heurística baseada na estratégia de Johnson (Johnson, 1974), a qual foi apresentada na Subseção 3.3.5 do Capítulo 3. Esta heurística, apesar de simplificar o algoritmo de discretização baseado em rough sets e lógica booleana, apresenta custo computacional $O(kn^3)$ para encontrar cada corte c e alocação de memória na ordem de $O(kn^2)$, onde k é o número de atributos e n o número de elementos ou linhas. Devido às restrições de memória impostas pela heurística e a capacidade disponível, somente 10% dos vetores poderiam ser discretizados. Portanto, a heurística de discretização não foi utilizada para os dados em questão.

Com o propósito de alcançar uma discretização satisfatória dos atributos, implementou-se também um algoritmo baseado na densidade de probabilidade do atributo. Este algoritmo toma como entrada um vetor N (com os valores de atributo) e o número c de cortes ou faixas de discretização. Inicialmente, o algoritmo computa o histograma de N (do atributo) considerando somente seus valores distintos D, ordenados crescentemente. A densidade encontrada para cada valor de D é sucessivamente somada e acumulada em D', de tal sorte que $D'(d) = D(d) + D'(d-1)$. Posteriormente, o vetor D' é dividido em c pedaços de mesmo tamanho, sendo C o vetor de índices que determinam a divisão de D'. Os valores de D' referenciados pelos

Tabela 6.2: Conjunto de atributos categóricos disponíveis para compor a Tabela de Decisão.

Nº	**Atributos**	**Valores Distintos**	**Tipo**	**Distribuição**
1	Id	40.492	String	Categórico
2	$Resultado\_Str$	NORMAL ou FRAUDE	String	Categórico
3	$Resultado\_Num$	2	Numérico	Categórico
4	$Atividade$	449	Numérico	Categórico
5	$Classe$	8	Numérico	Categórico
6	$Tipo\_Lig$	4	Numérico	Categórico
7	$Municipio$	72	Numérico	Categórico
8	$Media\_Consumo\_Discret$	10	Numérico	Categórico
9	$Dp\_Consumo\_Discret$	10	Numérico	Categórico
10	$Delta\_Consumo\_Discret$	10	Numérico	Categórico
11	$Media\_Trafo\_Discret$	10	Numérico	Categórico
12	$Dp\_Trafo\_Discret$	10	Numérico	Categórico

índices em C definem as faixas de discretização de N. Ao final, basta identificar a quais faixas pertencem cada valor em N, alcançando um novo vetor N' contendo c valores distintos.

Utilizando o algoritmo descrito acima, os atributos da Tabela 6.1 foram discretizados em 10 valores ou classes. O número de classes ou faixas de discretização interfere na generalização (especificação) do atributo. Portanto, a discretização dos atributos em 10 classes não é uma regra desta etapa da metodologia.

Os nomes dos atributos discretizados receberam o acréscimo do termo $Discret$. Por exemplo, o nome do atributo $Media\_Consumo$ foi modificado para $Media\_Consumo\_Discret$.

6.2.2 Seleção de Atributos

A composição de uma Tabela de Decisão depende de uma das etapas mais importantes do processo de descoberta de conhecimento em banco de dados: a seleção de atributos. A Tabela 6.2 ilustra o conjunto de atributos categóricos disponíveis para compor a Tabela de Decisão.

Como visto nos Capítulos 3 e 4, seja A o conjunto de atributos condicionais de uma Tabela de Decisão, um reduto $P \subseteq A$ é um subconjunto de atributos que mantém as relações de indiscernibilidade definidas por A. Ou seja, se P tem cardinalidade menor ou igual a A e consegue manter a mesma representação dos exemplos de uma dada Tabela de Decisão, então P é um reduto de A. Através de uma matriz de discernimento, apresentada na Subseção 4.6.2 do Capítulo 4, encontra-se o reduto de menor cardinalidade para qualquer Tabela de Decisão, chamado reduto ótimo.

Porém, para se construir uma matriz de discernimento necessita-se uma alocação de memória da ordem de $O(kn^2/2)$, onde k é o número de atributos condicionais e n o número de elementos ou linhas. Para a Tabela de Decisão em questão (não considerando Id, $Resultado\_Str$ e

$Municipio$), seriam necessários $9 * 40.492^2/2 = 7.378,209.288$ bytes ou, aproximadamente, 7 Gbytes de memória. Como a memória disponível é de 1 Gbyte, não foi possível encontrar o reduto ótimo pela construção de uma matriz de discernimento.

Apesar da metodologia proposta utilizar uma Tabela de Decisão com atributos bem definidos, a mesma deve ser aplicada considerando-se diferentes conjuntos de atributos, ainda que seja possível computar e identificar o reduto ótimo. Esta estratégia é necessária pois não se sabe previamente a qualidade das informações contidas nos atributos. É possível, portanto, que um atributo de um reduto ótimo contenha informações prejudiciais à descoberta de padrões de comportamento fraudulentos. Aplicando-se a metodologia à vários conjuntos de atributos, tem-se a possibilidade de alcançar melhores resultados para as medidas de avaliação consideradas.

6.2.3 Divisão Aleatória dos Dados para Treinamento e Teste

Uma das fases típicas do Aprendizado de Máquina é a divisão aleatória dos dados para treinamento e teste, como foi dito na Seção 2.3. O conjunto de treinamento consiste dos dados que serão submetidos à tratamentos e algoritmos de mineração, com o intuito de descobrir o conhecimento implícito. Já o conjunto de teste é utilizado para validar o treinamento, ou seja, avaliar o quanto o treinamento é representativo.

Nesta metodologia, optou-se por uma divisão igualitária entre os dados de treinamento e teste. Tomou-se, aleatoriamente, 20.246 linhas da Tabela de Decisão para formar o conjunto de treinamento e outros 20.246 restantes formaram o conjunto de teste.

A divisão dos dados pode ser feita considerando-se outras proporções, como por exemplo, 70% para treinamento e 30% para teste. Porém, acredita-se que a divisão igualitária promove uma melhor generalização dos dados de treinamento, evitando a sobreposição (ou *overfitting*) (Ng, 1997).

Até a etapa final da metodologia, onde o conjunto de teste é retomado, somente o conjunto de treinamento foi utilizado.

6.2.4 Operação *Unique*

A partir da Tabela de Decisão contendo apenas os registros de treinamento, realiza-se a operação *unique*, ou seja, identificar entre os 20.246 registros quais são distintos entre si. Esta operação compara cada par de registros possível, buscando aqueles que são idênticos. Quando encontra esta igualdade, elimina o segundo registro do par e atualiza um contador do primeiro registro. Ao final desta operação, tem-se todos os registros distintos, cada qual contendo um contador que informa quantas ocorrências do registro havia na Tabela de Decisão. Este contador recebe o nome de *suporte* e está representado por um vetor com o número de elementos igual ao número de registros distintos remanescentes. Note que o suporte informa o quanto o

conhecimento contido no registro é relevante, uma vez que quanto maior for o seu valor, maior é a ocorrência do mesmo na Tabela de Decisão.

O custo computacional da operação *unique* é da ordem de $O(n^2)$, ocupando $O(kn)$ bytes de memória, onde k é o número de atributos e n o número de elementos ou linhas.

6.2.5 Operação Aproximações

O estado corrente da Tabela de Decisão pode ser formalmente definido por $T = (U, C, d)$, em que: U é o conjunto de registros distintos e doravante chamados *padrões*, os quais foram encontrados na Subseção 6.2.4; C é o conjunto de atributos condicionais selecionados na Subseção 6.2.2; d é o atributo de decisão.

Os padrões em U possuem os valores 1 ou 2 para o atributo d, conforme seus resultados de inspeção sejam normal ou fraude, respectivamente. Sendo assim, pode-se distinguir em U o subconjunto de padrões normais $N \subset U$ e o subconjunto de padrões fraudulentos $F \subset U$, onde $|N| + |F| = |U|$. Os subconjuntos N e F, portanto, representam os *conceitos* de padrões normais e fraudulentos.

A operação *aproximações* encontra os conjuntos $\underline{C}N$, $\underline{C}F$, $BN_C(N)$ e $BN_C(F)$. Todos os padrões pertencentes ao conjunto $\underline{C}N$ são classificados como normais e com certeza não existe nenhum outro padrão com os mesmos valores de atributos condicionais e classificado como fraudador. Da mesma forma, todos os padrões pertencentes ao conjunto $\underline{C}F$ são classificados como fraudadores e com certeza não existe nenhum outro padrão com os mesmos valores de atributos condicionais e classificado como normal.

Os conjuntos $BN_C(N)$ e $BN_C(F)$ têm a mesma cardinalidade, pois para todo padrão em $BN_C(N)$ há um outro padrão semelhante em $BN_C(F)$, porém com resultado distinto. Devido a esta relação entre os padrões das duas regiões de fronteira, a operação aproximações cria uma outra tabela chamada $neighbor\_rate$. As linhas desta tabela armazenam informações referentes aos pares de padrões semelhantes em $BN_C(N)$ e $BN_C(F)$. Estas informações são:

Coluna1 índice (ou número da linha em T) do padrão normal;

Coluna2 índice (ou número da linha em T) do padrão fraudador;

Coluna3 valor de suporte do padrão normal;

Coluna4 valor de suporte do padrão fraudador;

Coluna5 valor da razão $Coluna3/Coluna4$.

Com a tabela $neighbor\_rate$, os padrões que estão nas regiões de fronteira $BN_C(N)$ e $BN_C(F)$ podem ser tratados como um único padrão, chamado de *padrão de fronteira*. Quanto menor o valor da Coluna5 que relaciona os suportes normal e fraudador, maior é o "caráter

fraudulento" do padrão de fronteira. Analogamente, quanto maior o valor da Coluna5, maior é o "caráter normal" do padrão de fronteira. A Tabela 6.3 ilustra o modelo de uma tabela $neighbor\_rate$ qualquer.

Tabela 6.3: Modelo de uma tabela $neighbor\_rate$ qualquer.

Coluna1	Coluna2	Coluna3	Coluna4	Coluna5
1.346	3	7	2	3.5
544	6	9	2	4.5
1.267	44	18	3	6
436	51	25	3	8.33
1.490	73	8	1	8
22	221	153	21	7.3
1885	241	2	3	0.67
42	279	13	1	13
871	305	1	3	0.33
2310	306	3	2	1.5

6.2.6 Operação Cortes

Após a operação aproximações, os padrões fraudulentos estarão contidos em $\underline{C}F$ ou em $BN_C(F)$ (o mesmo ocorre para os padrões normais). A cardinalidade destes conjuntos depende da divisão aleatória dos dados e principalmente da quantidade de atributos considerados na Tabela de Decisão. Isto porque, quando há poucos atributos, a chance de um padrão fraudulento ser semelhante a um padrão normal tende a ser maior. Em contrapartida, esta semelhança tende a diminuir quando se tem muitos atributos para distinguir entre normais e fraudadores.

A operação *cortes* toma como entrada a tabela $neighbor\_rate$, construída na Subseção 6.2.5, e ordena de forma crescente suas linhas de acordo com o valor da Coluna5, também chamada *razão*, a qual relaciona os suportes dos padrões normal e fraudador. Após a ordenação, a primeira linha de $neighbor\_rate$ terá o padrão de fronteira com maior "caráter fraudulento", enquanto a última terá o padrão com maior "caráter normal". Cada valor distinto da Coluna5 é chamado de *ponto de corte*, ou simplesmente *corte*, e representa um possível ponto de separação entre os padrões de fronteira de "caráter fraudulento" e "caráter normal". Em seguida, a operação cortes cria uma nova tabela chamada $neighbor\_rate\_distinct$ ou nrd, na qual há uma linha para cada corte. As colunas da tabela nrd são:

Coluna1: valor do corte;

Coluna2: somatório dos suportes dos padrões fraudulentos em $\underline{C}F$;

Coluna3: somatório dos suportes dos padrões fraudulentos em $neighbor\_rate$ que possuem razão menor ou igual ao valor de corte. Ou seja, o número de registros fraudulentos que estão na fronteira com "caráter fraudulento" em relação ao corte considerado;

Coluna4: somatório dos suportes dos padrões fraudulentos em $neighbor\_rate$ que possuem razão maior que o valor de corte. Ou seja, o número de registros fraudulentos que estão na fronteira com "caráter normal" em relação ao corte considerado;

Coluna5: somatório dos suportes dos padrões normais em $neighbor\_rate$ que possuem razão menor ou igual ao valor de corte. Ou seja, o número de registros normais que estão na fronteira com "caráter fraudulento" em relação ao corte considerado;

Coluna6: somatório dos suportes dos padrões normais em $neighbor\_rate$ que possuem razão maior que o valor de corte. Ou seja, o número de registros normais que estão na fronteira com "caráter normal" em relação ao corte considerado;

Coluna7: resultado da expressão $(Coluna2 + Coluna3)/(Coluna2 + Coluna3 + Coluna5)$.

As linhas da tabela nrd apresentam uma avaliação quantitativa dos padrões de fronteira encontrados pela metodologia. Em especial, a Coluna7 informa qual seria o "rendimento" obtido ao considerar os padrões de fraude da fronteira definidos pelo corte como sendo estritamente padrões fraudulentos (pertencentes à $\underline{C}F$). Quanto menor o valor do corte, maior é o "caráter fraudulento" dos padrões de fronteira englobados por este corte. Porém, devido a grande diferença na quantidade de registros normais e fraudulentos, um corte de valor muito pequeno engloba poucos padrões de fronteira. Da mesma forma, quanto maior o valor de corte, maior é a quantidade de padrões de fronteira com "caráter normal" que são considerados padrões fraudulentos.

6.2.7 Avaliação e Escolha do Corte

A etapa final da metodologia consiste em selecionar um conjunto de padrões fraudulentos, chamados *padrões finais*, gerar uma regra para cada padrão selecionado e testá-las no conjunto de teste. Incondicionalmente, fazem parte dos padrões finais os elementos que formam o conjunto $\underline{C}F$. Os demais padrões fraudulentos são definidos justamente pelo ponto de corte escolhido, separando aqueles padrões de fronteira com "caráter fraudulento" suficiente para compor o conjunto de padrões finais.

Como visto no final da Seção 6.2.6, a escolha do ponto de corte não é uma tarefa trivial e determinística. Sendo assim, esta metodologia propõe a inserção gradual de padrões fraudulentos de fronteira no conjunto de padrões finais e o sucessivo teste das regras geradas.

Portanto, a partir dos padrões em $\underline{C}F$ e do conjunto de padrões fraudulentos da fronteira que possuem razão menor ou igual ao valor de corte da primeira linha de nrd, define-se o

primeiro conjunto de padrões finais. Gera-se, então, um conjunto de regras à partir destes padrões finais e testa-se estas regras no conjunto de teste. Em uma outra iteração, gera-se o segundo conjunto de padrões finais, agora contendo $\underline{C}F$ e os padrões fraudulentos da fronteira que possuem razão menor ou igual ao valor de corte da segunda linha de nrd. Da mesma forma, gera-se um novo conjunto de regras e aplica-se as mesmas ao conjunto de teste. Obviamente, os padrões fraudulentos de fronteira contidos no primeiro conjunto de padrões finais também estarão contidos no segundo conjunto de padrões finais. O processo é repetido para cada linha de nrd, sendo que na última etapa todos os padrões fraudulentos de fronteira farão parte dos padrões finais.

Medidas de Avaliação

O teste de cada conjunto de regras gera quatro valores, de acordo com os acertos e erros de classificação:

Verdadeiros Positivos (VP): quantidade de registros de teste classificados corretamente como fraudulentos;

Falsos Positivos (FP): quantidade de registros de teste classificados erroneamente como fraudulentos;

Verdadeiros Negativos (VN): quantidade de registros de teste classificados corretamente como normais;

Falsos Negativos (FN): quantidade de registros de teste classificados erroneamente como normais.

A partir dos valores acima, definem-se as *medidas de avaliação* consideradas na escolha do melhor conjunto de regras e, conseqüentemente, do ponto de corte adequado. Estas medidas são:

Taxa de Acerto de Fraudes (TAF): quantidade de classificações fraudulentas corretas pelo total de classificações fraudulentas efetuadas. Esta medida é calculada por $VP/(VP + FP)$;

Fraudes Detectadas (FD): quantidade de classificações fraudulentas corretas pelo total de registros de teste com resultado fraude (TF). Esta medida é calculada por VP/TF;

Número de Inspeções (NI): total de classificações fraudulentas efetuadas, independente de seu resultado. Esta medida é calculada por $VP + FP$.

Curvas de Resultados

Para uma análise conjunta das 3 medidas de avaliação, plotam-se os resultados encontrados para os padrões em $\underline{C}F$ e para cada conjunto de padrões finais. As curvas geradas para cada medida de avaliação também são nomeadas por TAF (Taxa de Acerto de Fraudes), FD (Fraudes Detectadas) e NI (Número de Inspeções). O eixo das abcissas é composto por valores entre 0 e n, onde n é o número de conjuntos de padrões finais. A abscissa 0 indexa os valores das medidas de avaliação obtidos somente pelos padrões em $\underline{C}F$, enquanto as demais abscissas $(1, 2, ..., n)$ indexam os valores obtidos para cada conjunto de padrões finais. O eixo das ordenadas tem valores reais no intervalo de 0 a 1, representando o resultado das medidas de avaliação.

Para que a curva NI seja visualizada juntamente com TAF e FD, seus valores são divididos pelo número de registros de teste (no caso 20.246), de forma à pertencerem ao intervalo [0,1]. Sendo assim, NI torna-se a porcentagem de registros de testes inspecionados.

Escolha do Corte

De posse das curvas TAF, FD e NI, tem-se a possibilidade de escolher o conjunto de regras (o corte) que melhor satisfaz as expectativas de desempenho, parametrizando o processo de detecção de fraudes. Por exemplo, para encontrar o conjunto de regras que propicia uma taxa de acerto de 30%, busca-se pelo ponto em TAF com ordenada 0,3. A abscissa deste ponto informa o corte (o conjunto de regras) e indexa a quantidade de registros de teste classificados como fraude (NI) e a porcentagem de fraudes detectadas no conjunto de teste (FD). Caso deseja-se encontrar um conjunto de regras que leve a uma porcentagem de fraudes detectadas de 50%, por exemplo, busca-se pelo ponto em FD com ordenada 0,5. A abscissa deste ponto informa o conjunto de regras e indexa a quantidade de registros de teste classificados como fraude (NI) e a taxa de acerto de fraudes (TAF).

Também é possível definir o conjunto de regras a partir da quantidade de inspeções que se deseja realizar. Por exemplo, caso deseja-se inspecionar 5.000 clientes de uma localidade, submete-se os mesmos a cada conjunto de regras disponível. Encontrado o conjunto de regras que classifica como fraude todos os 5.000 clientes, identifica-se a abscissa corresponde e os valores esperados para TAF e FD.

6.3 Teste de Confiabilidade da Metodologia

Como visto na Subseção 6.2.3, após a discretização e a seleção dos atributos relevantes, divide-se os dados aleatoriamente em duas partes iguais: os conjuntos de treinamento e teste. A princípio, selecionar de forma aleatória a metade dos registros para compor o conjunto de treinamento não garante que os padrões encontrados posteriormente representem todos os da-

dos. Ou seja, não se tem certeza que os registros de teste possuirão um registro semelhante no conjunto de treinamento, possibilitando uma classificação correta na etapa de teste das regras. Possivelmente, os registros semelhantes podem ser selecionados para compor somente o conjunto de teste, comprometendo a identificação de padrões de treinamento e os resultados da etapa final de teste das regras.

Apesar das considerações acima, a metodologia para detecção de fraudes mostrou-se confiável. Para ilustrar sua aplicação aos dados disponíveis e comprovar sua robustez, esta seção apresenta um teste de confiabilidade da metodologia proposta. Considerando a Tabela de Decisão contendo os atributos $Media\_Consumo\_Discret$, $Media\_Trafo\_Discret$, $Tipo\_Lig$ e $Dp\_Consumo\_Discret$, além do atributo de decisão $Resultado\_Num$, as subseções seguintes apresentam quatro testes da metodologia à partir da etapa de divisão aleatória dos dados.

6.3.1 Teste A

Considerando a Tabela de Decisão $T = (U, C, d)$ com os atributos enunciados acima, foram selecionados aleatoriamente os registros para compor o conjunto de treinamento do Teste A. Dentre os 20.246 registros selecionados, 19.314 (95,4%) possuíam resultado normal e 932 (4,6%) resultado fraude, ou seja, a mesma porcentagem de fraudadores encontrada nos 40.492 registros, como visto na Subseção 5.3.4.

Com a operação *unique*, foram encontrados 2.229 padrões normais e 488 padrões fraudulentos. Após a operação aproximações, a tabela $neighbor\_rate$ alcançou 467 linhas, representando os padrões de fronteira, enquanto os conjuntos $\underline{C}N$ e $\underline{C}F$ obtiveram cardinalidade 1.762 e 21, respectivamente. A tabela nrd apresentou 109 linhas que propiciaram a formação de conjuntos de padrões finais distintos, dos quais foram derivadas regras. A partir do teste de cada conjunto de regras, obteve-se os resultados das medidas de avaliação, com suas respectivas curvas representadas na da Figura 6.1.

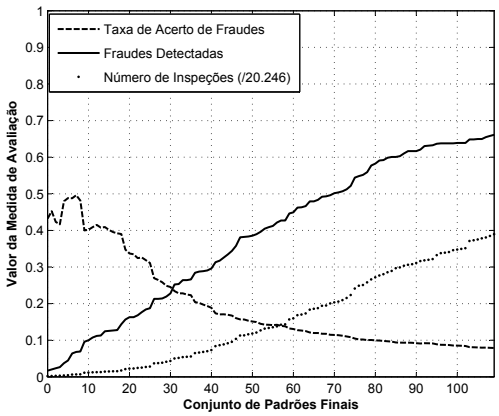

Figura 6.1: Resultado das medidas de avaliação TAF, FD e NI para Teste A

Observando as curvas da Figura 6.1, nota-se que o maior valor de TAF (0,5) está na abscissa 7, com o valor de FD em 0.07. O maior valor de FD (0,66) corresponde à abscissa 109, com o valor de TAF em 0,08. O ponto de intersecção entre as curvas apresenta abscissa 30,5 e ordenada 0,24. Os pontos da curva NI estão limitados à ordenada 0,4, indicando que, no pior caso, 40% dos clientes (registros) de teste são inspecionados.

6.3.2 Teste B

Novamente, foram selecionados aleatoriamente os registros para compor o conjunto de treinamento do Teste B, onde 19.336 (95,5%) possuíam resultado normal e 910 (4,5%) resultado fraude. Com a operação *unique*, foram encontrados 2.223 padrões normais e 488 padrões fraudulentos. A tabela $neighbor\_rate$ alcançou 457 linhas, enquanto os conjuntos $\underline{C}N$ e $\underline{C}F$ obtiveram cardinalidade 1.766 e 31, respectivamente. A tabela nrd apresentou 108 linhas, permitindo a formação dos conjuntos de padrões finais avaliados pelas curvas da Figura 6.2.

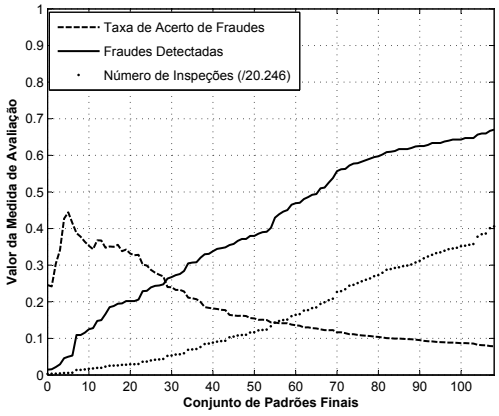

Figura 6.2: Resultado das medidas de avaliação TAF, FD e NI para Teste B.

Comparando as Figuras 6.1 e 6.2, vê-se que a partir da origem até a décima abscissa, aproximadamente, as curvas de TAF apresentam valores distintos. Porém, as curvas de FD para este mesmo intervalo apresentam valores menores ou iguais à 0,10, realizando um total de inspeções em torno de 2% ($NI = 0,02$). Sendo assim, esta discrepância só é relevante na escolha de cortes que privilegiam os valores máximos de TAF, desprezando a porcentagem de fraudes detectadas e inspecionando um número ínfimo de clientes.

Percorrendo as demais abscissas, em ordem crescente, vê-se que os valores das curvas dos Testes A e B são semelhantes.

6.3.3 Teste C

Neste Teste, foram selecionados 19.317 (95,4%) registros com resultado normal e 929 (4,6%) com resultado fraude. Após a busca por registros distintos, foram identificados 2.222 padrões normais e 508 fraudulentos. A tabela $neighbor\_rate$ contou com 488 linhas, enquanto os conjuntos $\underline{C}N$ e $\underline{C}F$ obtiveram cardinalidade 1.734 e 20, respectivamente. Já a tabela nrd apresentou 115 linhas, possibilitando a formação dos conjuntos de padrões finais avaliados pelas curvas da Figura 6.3.

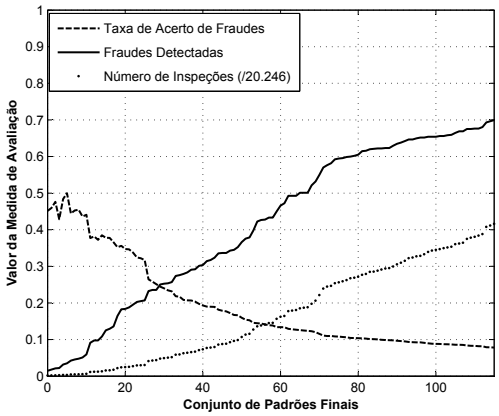

Figura 6.3: Resultado das medidas de avaliação TAF, FD e NI para Teste C.

A mesma análise do Teste B é válida para o Teste C: comparando as Figuras 6.1, 6.2 e 6.3, vê-se que os Testes A, B e C apresentam resultados semelhantes, com as curvas TAF, FD e NI mantendo os mesmos comportamentos gerais.

6.3.4 Teste D

No último Teste, foram selecionados 19.302 (95,3%) registros com resultado normal e 944 (4,7%) com resultado fraude. Posteriormente, foram identificados 2.228 padrões normais e 508 padrões fraudulentos. A tabela $neighbor\_rate$ apresentou 476 linhas, enquanto os conjuntos $\underline{C}N$ e $\underline{C}F$ contaram com 1.752 e 32 elementos, respectivamente. Finalmente, a tabela nrd apresentou 107 linhas, possibilitando a formação dos conjuntos de padrões finais avaliados pelas curvas da Figura 6.4.

Novamente, as curvas das Figura 6.4 são semelhantes com as demais apresentadas nas subseções anteriores.

6.3.5 Análise dos Testes

A quantidade de conjuntos de padrões finais depende do número de cortes distintos da tabela $neighbor\_rate$, ou seja, das linhas da tabela nrd. Os quatro Testes apresentados possuem

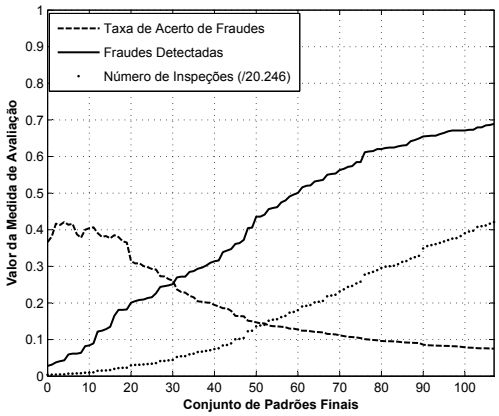

Figura 6.4: Resultado das medidas de avaliação TAF, FD e NI para Teste D.

quantidades diferentes de conjuntos de padrões finais, isto é, possuem tabelas nrd com número de linhas distintos. Além disso, os cortes encontrados em cada Teste não são necessariamente os mesmos, pois dependem da seleção aleatória de registros.

Apesar destas adversidades, é possível visualizar o comportamento geral das curvas TAF, FD e NI através da média entre os valores apresentados em cada Teste. Devido à diferença na quantidade de cortes, as abscissas foram limitadas à 107, valor máximo comum às quatro tabelas nrd. A Figura 6.5 ilustra em preto destacado as curvas médias TAF, FD e NI e em cinza as mesmas curvas encontradas nos Testes (1 a 4), evidenciando a semelhança de comportamento. Portanto, quaisquer que sejam os registros aleatórios que componham um conjunto de treinamento, as curvas das medidas de avaliação tentem à apresentar um comportamento comum, testificando a confiabilidade da metodologia proposta.

6.4 Avaliação de Conjuntos de Atributos

Com o intuito de encontrar as características que proporcionam os melhores desempenhos na detecção de fraudes, esta seção apresenta a aplicação (ou teste) da metodologia à alguns conjuntos de atributos condicionais, organizados pelo número de atributos considerados.

Como dito na Subseção 6.2.2, não foi possível computar o reduto devido à grande quantidade de registros e a limitada disponibilidade de memória. Desta forma, a estratégia de testar alguns conjuntos de atributos condicionais não garante que o reduto seja avaliado. Porém, devido à abrangência das avaliações e a comparação de seus resultados, evidencia-se que os atributos mais qualificados para o problema foram analisados.

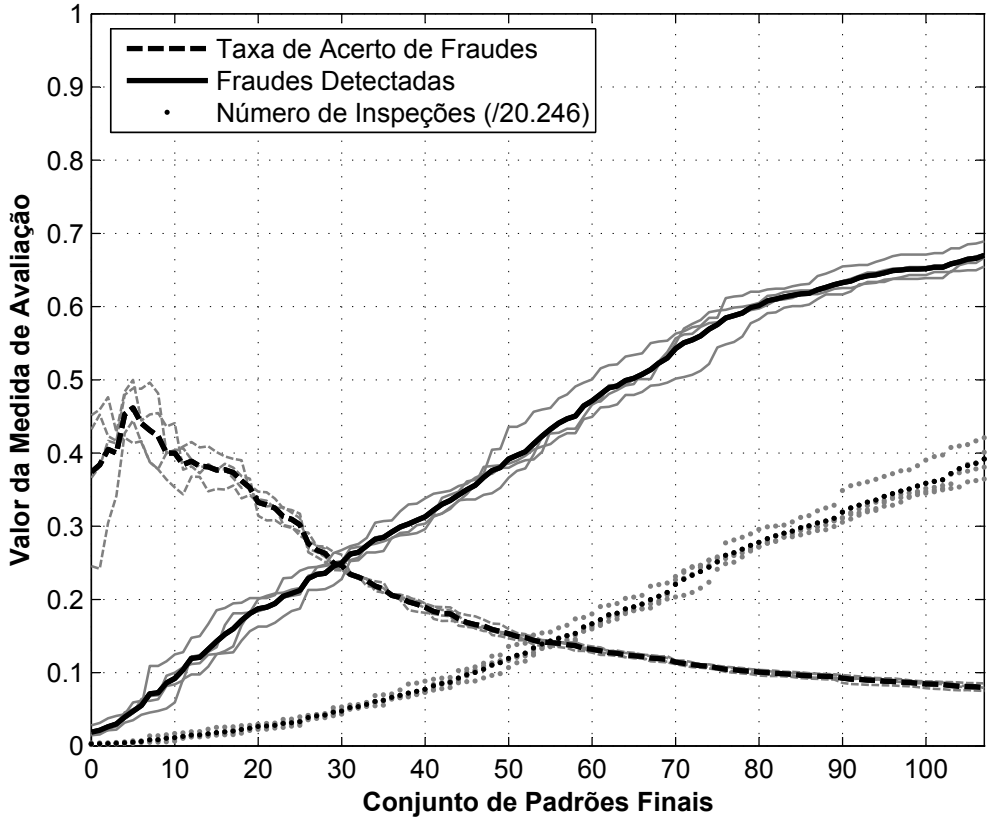

Figura 6.5: Curvas médias TAF, FD e NI.

6.4.1 Conjuntos com 2 Atributos

Tomando 2 atributos condicionais dentre os disponíveis na Tabela 6.2, foram avaliados os 5 conjuntos de atributos enunciados abaixo.

Conjunto 2.1

$$\mathbf{1 - Media\_Consumo\_Discret}$$
$$\mathbf{2 - Dp\_Consumo\_Discret}$$

Considerando os atributos condicionais acima, o Conjunto 2.1 apresentou os resultados ilustrados na Figura 6.6.

Analisando os resultados da Figura 6.6 na ordem crescente das abscissas, vê-se que a TAF apresentou crescimento até o terceiro conjunto de padrões finais, estabilizou-se até o sexto e decresceu até seu final. Isto mostra que somente os 3 primeiros conjuntos de padrões finais apresentam forte caráter fraudulento, ou seja, são formados por padrões que deveriam pertencer à $\underline{C}F$. A partir do sexto conjunto de padrões finais, a curva TAF sofre reduções ou mantém-se

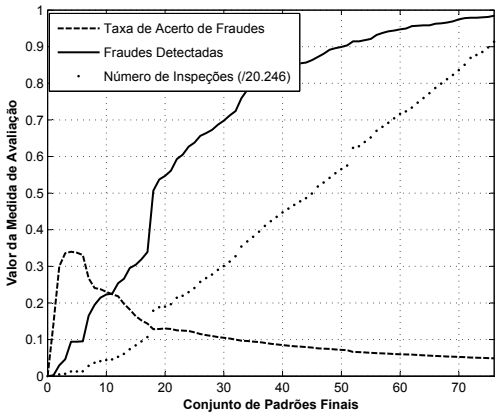

Figura 6.6: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 2.1.

constante, porém a FD tem comportamento praticamente inverso, marcando o início do comprometimento entre as duas medidas de avaliação. Quanto maior o número de regras geradas, maior é a quantidade de fraudes detectadas, porém ao custo de muitas inspeções realizadas com baixas taxas de acerto.

É interessante notar que a curva FD não sofre variação negativa. O incremento do número de regras para cada conjunto de padrões finais (em ordem crescente) só pode aumentar a quantidade de fraudes detectadas. Uma regra que leva a um Verdadeiro Positivo não muda sua atuação com a inserção de novas regras.

Conjunto 2.2

1 – Media_Consumo_Discret
2 – Tipo_Lig

Considerando os atributos condicionais acima, o Conjunto 2.2 apresentou os resultados ilustrados na Figura 6.7.

Comparando os resultados das Figuras 6.6 e 6.7, nota-se que o Conjunto 2.2 possui menor discernimento que o Conjunto 2.1 devido ao atributo $Tipo\_Lig$, o qual possui apenas 4 valores distintos. O desempenho das curvas TAF e FD também foi menor, sendo que a primeira não apresentou valores acima de 0,20. A grande diferença nos resultados dos Conjuntos 2.1 e 2.2 pode ser explicada pelas informações complementares contidas nos atributos condicionais $Media\_Consumo\_Discret$ e $Dp\_Consumo\_Discret$. A informação da média deve estar acompanhada do desvio-padrão para a completude do conhecimento acerca do comportamento de consumo dos clientes. Considerando os atributos $Media\_Consumo\_Discret$ e $Tipo\_Lig$, o Conjunto 2.2 deixa de conter um conhecimento completo do consumo dos registros selecionados, comprometendo os resultados das medidas de avaliação.

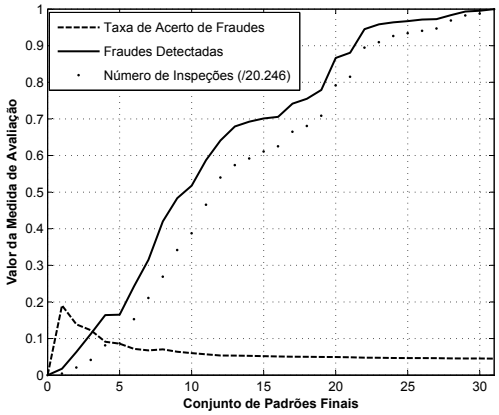

Figura 6.7: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 2.2.

Conjunto 2.3

$$\mathbf{1 - Media\_Consumo\_Discret}$$
$$\mathbf{2 - Classe}$$

Considerando os atributos condicionais acima, o Conjunto 2.3 apresentou os resultados ilustrados na Figura 6.8.

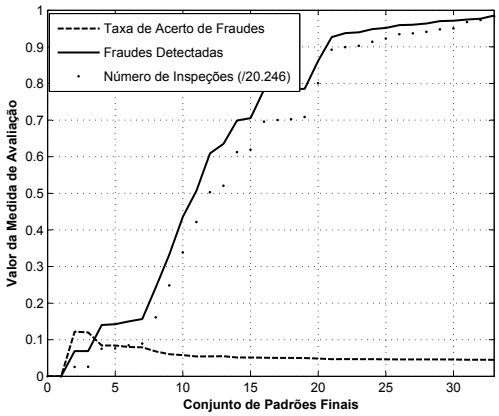

Figura 6.8: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 2.3.

Analisando os resultados das Figuras 6.7 e 6.8, nota-se que os Conjuntos 2.2 e 2.3 apresentaram comportamentos semelhantes, a despeito da quantidade de conjuntos de padrões finais e do máximo global das curvas TAF. A justificativa para o baixo desempenho do Conjunto 2.3 é a mesma apresentada para o Conjunto 2.2: a consideração da média de consumo sem o complemento do desvio-padrão.

Conjunto 2.4

$$\mathbf{1 - Classe}$$
$$\mathbf{2 - Tipo\_Lig}$$

Considerando os atributos condicionais acima, o Conjunto 2.4 apresentou os resultados ilustrados na Figura 6.9.

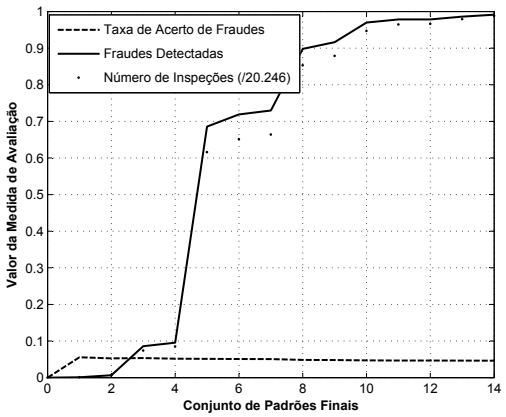

Figura 6.9: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 2.4.

Utilizando-se somente de atributos estáticos, o Conjunto 2.4 apresentou resultados inferiores aos demais conjuntos da Subseção 6.4.1, com valores de TAF limitados à 0,05. Partindo do princípio de que a classe e o tipo de ligação dos clientes não mudam com o passar do tempo, somente estes conhecimentos não são suficientes para indicar possíveis fraudadores. Porém, os atributos condicionais $Classe$ e $Tipo\_Lig$ têm sua contribuição no complemento do conhecimento fornecido pelos atributos dinâmicos.

Conjunto 2.5

$$\mathbf{1 - Media\_Consumo\_Discret}$$
$$\mathbf{2 - Delta\_Consumo\_Discret}$$

Considerando os atributos condicionais acima, o Conjunto 2.5 apresentou os resultados ilustrados na Figura 6.10.

O Conjunto 2.5 também considera 2 atributos condicionais discretizados em 10 classes distintas, como o Conjunto 2.1. Comparando as Figuras 6.6 e 6.10, vê-se que as curvas TAF e FD são semelhantes em muitas abscissas. Sendo assim, os Conjuntos 2.1 e 2.5 são indicados a serem acrescidos de outros atributos, tanto estáticos quanto dinâmicos.

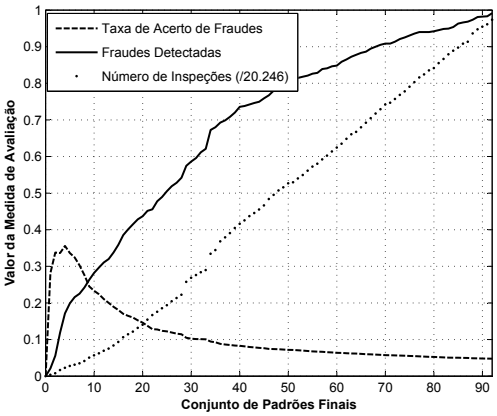

Figura 6.10: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 2.5.

6.4.2 Conjuntos com 3 Atributos

Considerando 3 atributos condicionais dentre os disponíveis na Tabela 6.2, foram avaliados 6 conjuntos de atributos, os quais são apresentados abaixo.

Conjunto 3.1

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Tipo_Lig

Considerando os atributos condicionais acima, o Conjunto 3.1 apresentou os resultados ilustrados na Figura 6.11.

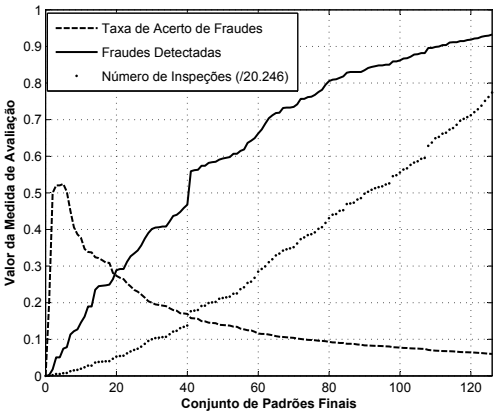

Figura 6.11: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 3.1.

O Conjunto 3.1 representa o Conjunto 2.1, da Subseção 6.4.1, acrescido de $Tipo\_Lig$. Comparando os resultados destes 2 conjuntos, vê-se que o incremento do atributo $Tipo\_Lig$ aumentou o discernimento entre os registros e proporcionou um melhor desempenho para a curva TAF, principalmente para as 10 primeiras abscissas. O ponto de intersecção, por exemplo, aumentou sua ordenada 5% (ao avançar de 0,23 a 0,28). Sendo assim, o acréscimo de $Tipo\_Lig$ proporcionou um complemento ao conhecimento dos atributos $Media\_Consumo\_Discret$ e $Dp\_Consumo\_Discret$ no sentido de identificar padrões de comportamento fraudulento.

Conjunto 3.2

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Classe

Considerando os atributos condicionais acima, o Conjunto 3.2 apresentou os resultados ilustrados na Figura 6.12.

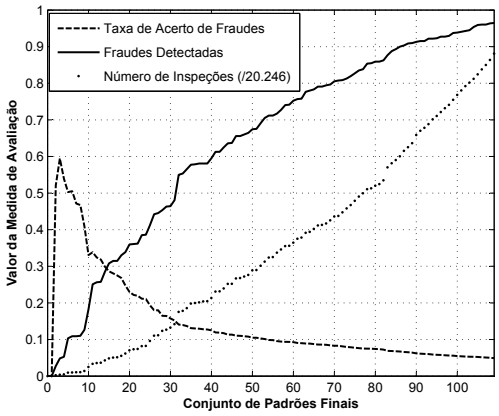

Figura 6.12: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 3.2.

Comparando as Figuras 6.11 e 6.12, é fácil ver que as curvas TAF e FD dos Conjuntos 3.1 e 3.2 apresentam comportamentos praticamente idênticos, a despeito do máximo global da curva TAF do Conjunto 3.2, que alcançou uma abscissa 7% (de 0,52 para 0,59) maior. Como o atributo $Tipo\_Lig$ do Conjunto 3.1, o atributo $Classe$ do Conjunto 3.2 complementou o conhecimento de $Media\_Consumo\_Discret$ e $Dp\_Consumo\_Discret$.

Conjunto 3.3

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Atividade

Considerando os atributos condicionais acima, o Conjunto 3.3 apresentou os resultados ilustrados na Figura 6.13.

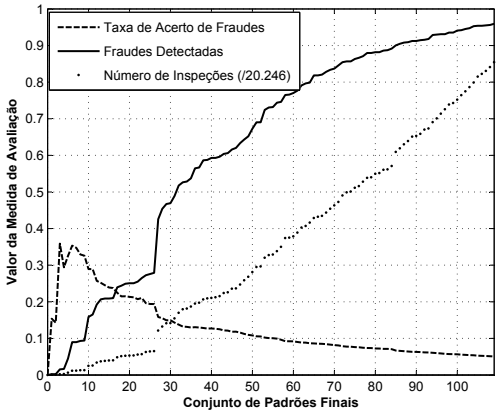

Figura 6.13: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 3.3.

O Conjunto 3.3 representa o Conjunto 2.1, da Subseção 6.4.1, acrescido de $Atividade$. Comparando os resultados destes 2 conjuntos, vê-se que o incremento do atributo $Atividade$ aumentou o discernimento entre os registros, porém não proporcionou um melhor desempenho para as curvas TAF e FD. Com o atributo $Atividade$, as curvas encontradas apresentaram muitas variações abruptas de uma abscissa para outra, ou seja, o atributo introduziu uma instabilidade. Portanto, o atributo $Atividade$ não complementa o conhecimento dos atributos $Media\_Consumo\_Discret$ e $Dp\_Consumo\_Discret$, pelo contrário, prejudica a continuidade das curvas de resultados.

Conjunto 3.4

1 – Media_Consumo_Discret
2 – Delta_Consumo_Discret
3 – Tipo_Lig

Considerando os atributos condicionais acima, o Conjunto 3.4 apresentou os resultados ilustrados na Figura 6.14.

Os Conjuntos 3.1 e 3.4 diferem-se, respectivamente, pelos atributos $Dp\_Consumo\_Discret$ e $Delta\_Consumo\_Discret$, permitindo avaliar qual deles apresenta melhor desempenho ao complementar $Media\_Consumo\_Discret$ e $Tipo\_Lig$. Comparando o comportamento das curvas TAF e FD das Figuras 6.11 e 6.14, vê-se que a primeira apresentou maiores ordenadas, principalmente no máximo global da curva TAF. Portanto o Conjunto 3.1, com o atributo $Dp\_Consumo\_Discret$, teve melhor desempenho que o Conjunto 3.4 na aplicação da metodologia.

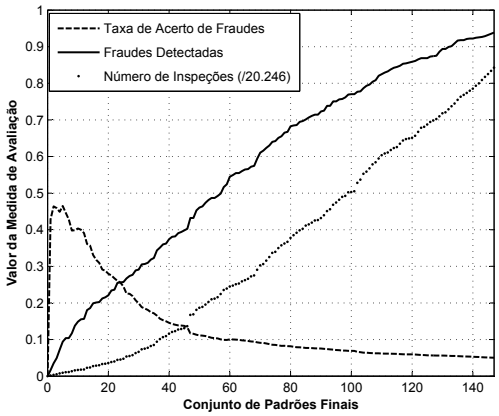

Figura 6.14: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 3.4.

Conjunto 3.5

1 – Media_Consumo_Discret
2 – Delta_Consumo_Discret
3 – Classe

Considerando os atributos condicionais acima, o Conjunto 3.5 apresentou os resultados ilustrados na Figura 6.15.

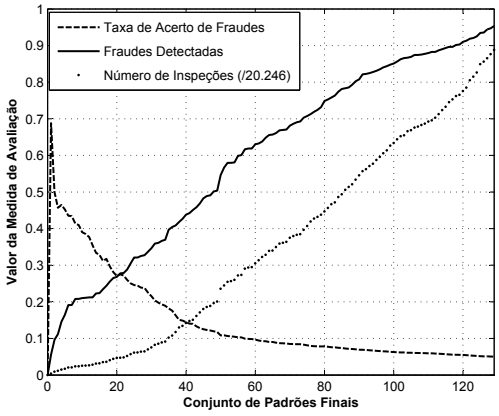

Figura 6.15: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 3.5.

Comparando os Conjuntos 3.2 e 3.5 através de suas respectivas Figuras 6.12 e 6.15, nota-se que as curvas TAF e FD apresentam algumas particularidades de desempenho que dificultam a definição do melhor conjunto. Por exemplo, ao buscar na Figura 6.12 uma TAF de 0,2, encontra-se o valor 0,41 para FD. Buscando pela mesma ordenada de TAF na Figura 6.15, encontra-se o valor 0,36 para FD. Portanto, o Conjunto 3.2 alcançou mais fraudes detectadas ao fixar a taxa de acerto em 0,2. Procedendo de forma inversa, ao buscar na Figura 6.12 a

ordenada 0,2 para FD, encontra-se o valor 0,33 para TAF. Buscando pela mesma ordenada de FD na Figura 6.15, encontra-se o valor 0,43 para TAF. Portanto, o Conjunto 3.5 alcançou uma maior taxa de acerto de fraudes ao fixar as fraudes detectadas em 0,2.

Conjunto 3.6

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Delta_Consumo_Discret

Considerando os atributos condicionais acima, o Conjunto 3.6 apresentou os resultados ilustrados na Figura 6.16.

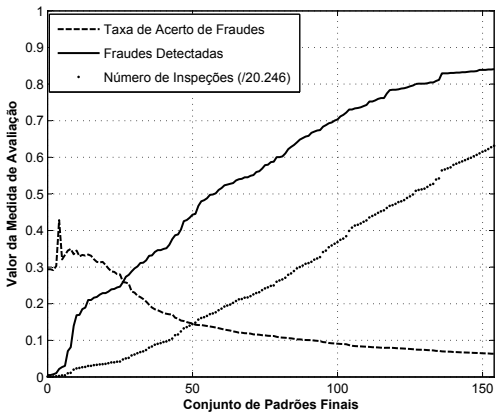

Figura 6.16: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 3.6.

O Conjunto 3.6 também representa extensões dos Conjuntos 2.1 e 2.5 da Subseção 6.4.1, pois considera os atributos $Dp\_Consumo\_Discret$ e $Delta\_Consumo\_Discret$ em conjunto com $Media\_Consumo\_Discret$. Como visto na Figura 6.16, os resultados das medidas de avaliação foram semelhantes aqueles alcançados pelos Conjuntos 3.4 e 3.5. Nota-se também que, por considerar somente atributos dinâmicos, apresentou o maior discernimento entre registros dentre os conjuntos com 3 atributos, contando com 154 cortes distintos.

Conjunto 3.7

1 – Media_Consumo_Discret
2 – Tipo_Lig
3 – Classe

Considerando os atributos condicionais acima, o Conjunto 3.7 apresentou os resultados ilustrados na Figura 6.17.

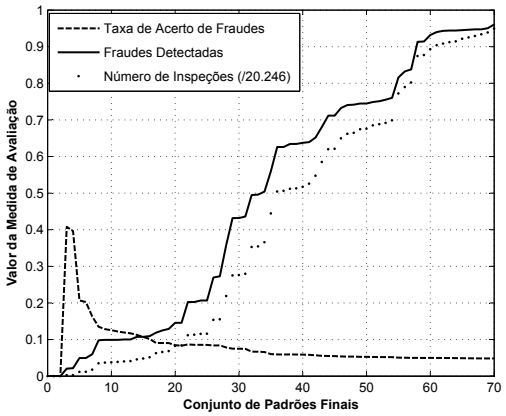

Figura 6.17: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 3.7.

O Conjunto 3.7 representa uma extensão dos Conjuntos 2.2 e 2.3 da Subseção 6.4.1. No entanto, à partir da comparação entre as Figuras 6.8, 6.9 e 6.17, nota-se que a utilização dos atributos $Classe$ e $Tipo\_Lig$ (e não somente um deles) juntamente com $Media\_Consumo\_Discret$ não proporciona melhorias nas medidas de avaliação. É possível, portanto, que o conhecimento dos atributos $Classe$ e $Tipo\_Lig$ sejam semelhantes, de tal sorte que a utilização de apenas um deles seja suficiente.

6.4.3 Conjuntos com 4 Atributos

Considerando 4 atributos condicionais dentre os disponíveis na Tabela 6.2, foram avaliados 4 conjuntos de atributos, os quais são apresentados abaixo.

Conjunto 4.1

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Classe
4 – Tipo_Lig

Considerando os atributos condicionais acima, o Conjunto 4.1 apresentou os resultados ilustrados na Figura 6.18.

O Conjunto 4.1 representa o Conjunto 3.2, da Subseção 6.4.2, acrescido de $Tipo\_Lig$. Comparando as Figuras 6.12 e 6.18 que ilustram os resultados dos Conjuntos 3.2 e 4.1, vê-se que as curvas TAF e FD apresentam comportamentos semelhantes. Porém, o incremento de $Tipo\_Lig$ aumentou o discernimento entre os registros, diminuindo os valores médios de NI. É interessante notar também que as curvas TAF dos Conjuntos 3.2 e 4.1 possuem orde-

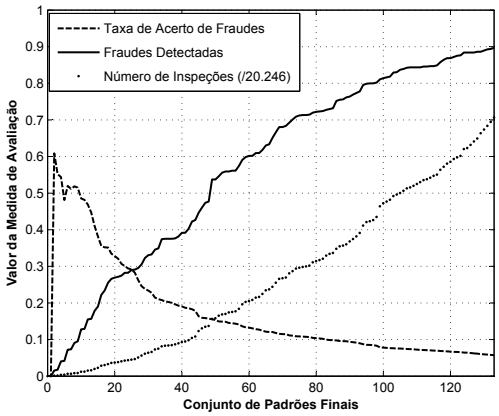

Figura 6.18: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 4.1.

nada 0,0 para a abscissa 0, significando que as regras derivadas à partir dos padrões de $\underline{CF}$ não realizaram nenhuma classificação fraudulenta de forma correta.

Conjunto 4.2

1 – Media_Consumo_Discret
2 – Delta_Consumo_Discret
3 – Classe
4 – Tipo_Lig

Considerando os atributos condicionais acima, o Conjunto 4.2 apresentou os resultados ilustrados na Figura 6.19.

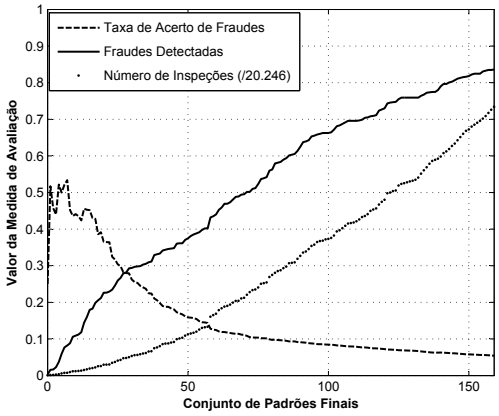

Figura 6.19: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 4.2.

Como na análise anterior, o Conjunto 4.2 representa o Conjunto 3.5, da Subseção 6.4.2, acrescido do atributo $Tipo\_Lig$. Comparando as Figuras 6.15 e 6.19 que ilustram os resul-

tados dos Conjuntos 3.5 e 4.2, vê-se que as curvas TAF e FD apresentam comportamentos semelhantes. Porém, o incremento de $Tipo\_Lig$ aumentou o discernimento entre os registros, diminuindo os valores médios de NI.

Conjunto 4.3

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Tipo_Lig
4 – Media_Trafo_Discret

Considerando os atributos condicionais acima, o Conjunto 4.3 apresentou os resultados ilustrados na Figura 6.20.

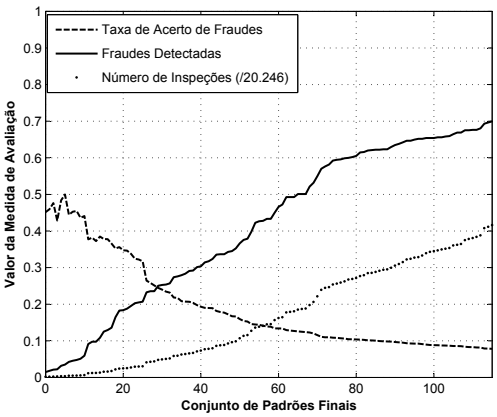

Figura 6.20: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 4.3.

O Conjunto 4.3 representa o Conjunto 3.1, da Subseção 6.4.2, com o acrescido do atributo $Media\_Trafo\_Discret$. Comparando os Conjuntos 3.1 e 4.3 pelos resultados das Figuras 6.11 e 6.20, vê-se que as curvas TAF e FD de cada conjunto apresentaram um comportamento distinto. O Acréscimo do atributo $Media\_Trafo\_Discret$ diminuiu os valores das curvas FD e NI, prejudicando o desempenho do Conjunto 4.3.

Conjunto 4.4

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Classe
4 – Media_Trafo_Discret

Considerando os atributos condicionais acima, o Conjunto 4.4 apresentou os resultados ilustrados na Figura 6.21.

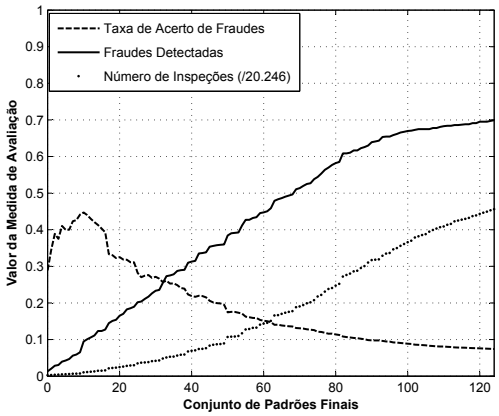

Figura 6.21: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 4.4.

Como na avaliação anterior, o Conjunto 4.4 representa o Conjunto 3.2, da Subseção 6.4.2, acrescido do atributo $Media\_Trafo\_Discret$. Comparando os Conjuntos 3.2 e 4.4 pelos resultados das Figuras 6.12 e 6.21, vê-se que as curvas TAF e FD de cada conjunto apresentaram um comportamento distinto. O Acréscimo do atributo $Media\_Trafo\_Discret$ diminuiu os valores das curvas FD e NI, prejudicando o desempenho do Conjunto 4.4.

6.4.4 Conjuntos com 5 Atributos

Considerando 5 atributos condicionais dentre os disponíveis na Tabela 6.2, foram avaliados 3 conjuntos de atributos, os quais são apresentados abaixo.

Conjunto 5.1

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Classe
4 – Tipo_Lig
5 – Delta_Consumo_Discret

Considerando os atributos condicionais acima, o Conjunto 5.1 apresentou os resultados ilustrados na Figura 6.22.

O Conjunto 5.1 representa o Conjunto 4.1, da Subseção 6.4.3, com o acréscimo do atributo $Delta\_Consumo\_Discret$. Porém, comparando os resultados dos Conjuntos 4.1 e 5.1 ilustrados em suas respectivas Figuras 6.18 e 6.22, vê-se que as curvas TAF e FD do segundo conjunto apresentaram comportamento inferior. O acréscimo de $Delta\_Consumo\_Discret$ foi desfavorável à descoberta de padrões fraudulentos, prejudicando o resultado das medidas de avaliação do Conjunto 5.1.

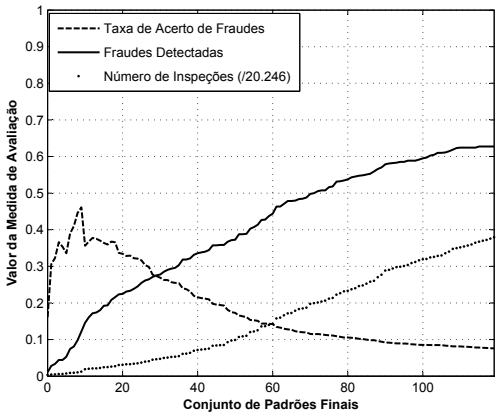

Figura 6.22: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 5.1.

Conjunto 5.2

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Classe
4 – Tipo_Lig
5 – Media_Trafo_Discret

Considerando os atributos condicionais acima, o Conjunto 5.2 apresentou os resultados ilustrados na Figura 6.23.

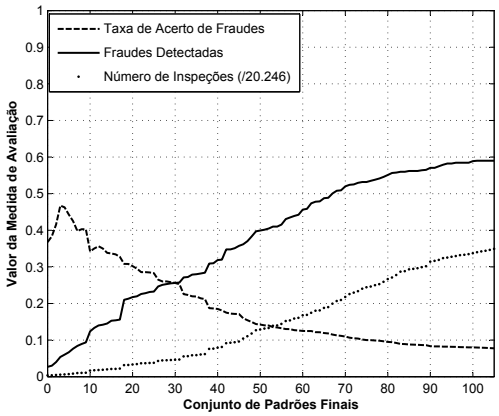

Figura 6.23: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 5.2.

O Conjunto 5.2 representa o Conjunto 4.1, da Subseção 6.4.3, com o acréscimo do atributo $Media\_Trafo\_Discret$. Porém, comparando os resultados dos Conjuntos 4.1 e 5.2 ilustrados em suas respectivas Figuras 6.18 e 6.23, vê-se que as curvas TAF e FD do segundo conjunto apresentaram comportamento inferior. O acréscimo do atributo $Media\_Trafo\_Discret$ foi

desfavorável à descoberta de padrões fraudulentos, prejudicando o resultado das medidas de avaliação do Conjunto 5.2.

Conjunto 5.3

1 – Media_Consumo_Discret
2 – Dp_Consumo_Discret
3 – Classe
4 – Media_Trafo_Discret
5 – Dp_Trafo_Discret

Considerando os atributos condicionais acima, o Conjunto 5.3 apresentou os resultados ilustrados na Figura 6.24. O ponto de intersecção entre as curvas TAF e FD tem ordenada 0,15 e abscissa 13, contabilizando 3.126 valores condicionais.

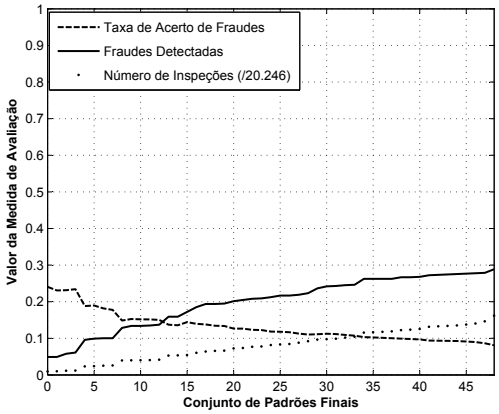

Figura 6.24: Resultado das medidas de avaliação TAF, FD e NI para o Conjunto 5.3.

O Conjunto 5.3 representa o Conjunto 3.1, da Subseção 6.4.2, acrescido de 2 atributos referentes ao Trafo: $Media\_Trafo\_Discret$ e $Dp\_Trafo\_Discret$. A comparação entre os resultados dos Conjuntos 3.1 e 5.3, ilustrados em suas respectivas Figuras 6.11 e 6.24, revela que o acréscimo dos atributos reduziu o desempenho das curvas TAF e FD, as quais não alcançaram valores acima de 0,3.

6.4.5 Resumo da Avaliação

A avaliação de conjuntos de atributos aponta as características de cliente mais indicadas para a detecção de fraude, como também aquelas que prejudicam o processo. A escolha do conjunto de atributos mais adequado deve levar em consideração alguns fatores:

- A qualidade dos atributos no banco de dados, ou seja, o quanto as informações do banco representam a realidade;

- A facilidade de acesso aos atributos, uma vez que alguns podem apresentar dificuldades de recuperação ou mesmo privilégios de acesso;
- A quantidade de valores distintos que o atributo possui e sua possibilidade de discretização em menores classes;
- A relação do atributo com o objetivo do problema.

Os conjuntos contendo $Media\_Consumo\_Discret$ e outros atributos estáticos apresentaram resultados inferiores aos alcançados por conjuntos com $Media\_Consumo\_Discret$ acompanhado por $Dp\_Consumo\_Discret$ e/ou $Delta\_Consumo\_Discret$. Isto pode ser explicado pela correlação de conhecimento entre os atributos de consumo, que é a informação fundamental sobre o comportamento dos clientes.

Dentre as avaliações realizadas, os atributos estáticos $Tipo\_Lig$ e $Classe$ complementaram com melhores resultados os conjuntos com atributos dinâmicos de consumo. Isto pode ser visto comparando o desempenho dos Conjuntos 4.1 e 4.2 com os demais avaliados.

Os valores de $Media\_Trafo\_Discret$ e $Dp\_Trafo\_Discret$ são calculados à partir das médias de consumo mensais em cada poste. Ou seja, o primeiro é uma "média de médias" e o segundo é o "desvio-padrão de médias". Portanto, a qualidade da informação destes dois atributos é inferior a dos demais atributos dinâmicos. Ao utilizar $Media\_Trafo\_Discret$ juntamente com $Dp\_Trafo\_Discret$ no Conjunto 5.3, o desempenho das medidas de avaliação foi comprometido.

O atributo $Atividade$ é muito utilizado pelas empresas de distribuição de energia elétrica para direcionar inspeções de clientes. Isto porque algumas atividades, principalmente comerciais e industriais, contabilizam muitos clientes fraudadores. O relacionamento entre atributos de consumo e de atividade de clientes, portanto, tenderia à melhorar os resultados das medidas de avaliação. Porém, incoerências detectadas entre a verdadeira atividade de alguns clientes e a atividade presente no banco de dados comprometeram os resultados encontrados com a utilização deste atributo.

Os atributos Id e $Municipio$ também não foram considerados pois suas informações não contribuem para a detecção de fraudes, servindo apenas para informar a identificação e o município dos clientes.

6.4.6 Estudo de Casos

Com a avaliação de conjuntos de atributos, ficam disponíveis vários conjuntos de regras, levando à diferentes valores para TAF e FD. Como dito na Subseção 6.2.7, a escolha do conjunto de regras a ser utilizado é baseada nos valores desejados de TAF e/ou FD. Sendo assim, esta seção apresenta o processo de escolha de 2 conjuntos de regras, aproveitando a avaliação de conjuntos de atributos já realizada.

Caso A

Considerando n unidades consumidoras da empresa de energia elétrica, deseja-se realizar um processo de inspeção que alcance uma taxa de acerto de fraude em torno de 30%. A Tabela 6.4 ilustra os maiores valores para FD e NI apresentados na avaliação de conjuntos de atributos, considerando $TAF = 0,3$.

Tabela 6.4: Melhores resultados buscados na avaliação de conjuntos de atributos.

Conjuntos	TAF	FD	NI
3.1	0,3	0,25	0,040
3.2	0,3	0,28	0,043
3.4	0,3	0,21	0,033
3.5	0,3	0,25	0,040
4.1	0,3	0,28	0,042
4.2	0,3	0,26	0,041

Como visto na Tabela 6.4, os Conjuntos 3.2 e 4.1 (que é complementar ao 3.2) apresentaram os maiores valores para FD: 0,28. Porém, o Conjunto 4.1 leva uma pequena vantagem na medida NI, realizando uma quantidade de inspeções menor que a do Conjunto 3.2 (0,042 e 0,043 respectivamente). Portanto, o Conjunto 4.1 (em seu corte 23) contém o conjunto de regras mais adequado para se alcançar uma taxa de acerto de fraude em torno de 30%.

Caso B

Considerando as mesmas n unidades consumidoras da empresa de energia elétrica, deseja-se realizar um processo de inspeção em que aproximadamente 40% dos clientes fraudadores sejam detectados. A Tabela 6.5 ilustra os maiores valores para TAF e NI apresentados na avaliação de conjuntos de atributos, considerando $FD = 0,4$.

Tabela 6.5: Melhores resultados buscados na avaliação de conjuntos de atributos.

Conjuntos	TAF	FD	NI
3.1	0,20	0,40	0,099
3.2	0,19	0,40	0,073
3.4	0,13	0,40	0,136
3.5	0,17	0,40	0,108
4.1	0,18	0,40	0,098
4.2	0,14	0,40	0,132

Vê-se na Tabela 6.5 que o Conjunto 3.1 apresentou o maior valor para TAF: 0,20. É interessante notar, no entanto, que o Conjunto 3.2, apesar de possuir um valor inferior de TAF

(0,19), apresentou o menor valor para NI: 0,073. Ou seja, o Conjunto 3.2 alcançou uma TAF muito próxima de 0,20, realizando um menor número de inspeções. Portanto, os Conjuntos 3.1 (em seu corte 30) e 3.2 (em seu corte 25) contém os conjuntos de regras mais adequados para se alcançar uma porcentagem de fraudes detectadas em torno de 40%.

6.5 Considerações Finais

Neste capítulo foi apresentada uma metodologia para detecção de fraudes usando Rough Sets. A partir dos vetores de atributos disponibilizados no Capítulo 5, foi aplicada a metodologia proposta, seguida por seu teste de confiabilidade. Atestada a confiabilidade da metodologia, vários conjuntos de atributos foram avaliados na busca pelas informações mais relevantes para a descoberta de padrões de comportamento fraudulento. Ao final, foram enunciados os atributos que beneficiaram e prejudicaram os resultados das medidas de avaliação consideradas.

No próximo capítulo são apresentadas as conclusões finais do trabalho, as contribuições alcançadas e os trabalhos futuros a serem realizados.

CAPÍTULO

7

Conclusão

7.1 Considerações Finais

Neste trabalho foi abordada a detecção de fraudes em unidades consumidoras de energia elétrica através da aplicação de uma metodologia baseada em conceitos de Rough Sets. O estudo aprofundado desta emergente técnica de Inteligência Artificial permitiu compreender sua atuação em dados organizados em Sistemas de Informação ou Tabelas de Decisão.

Ao aplicar alguns conceitos de Rough Sets e KDD aos dados de clientes consumidores de energia elétrica, foi possível analisar o relacionamento entre os padrões de comportamento normais e fraudulentos. A avaliação detalhada da região de fronteira entre estes padrões normais e fraudulentos é o ponto principal da metodologia proposta. Esta avaliação permite gerar vários conjuntos de regras que levam à fraude, cada qual focado em diferentes estimativas de taxa de acerto, quantidade de fraudes detectadas e número de inspeções. Portanto, o conjunto final de regras, simbolizando os comportamentos fraudulentos, é definido de acordo com o objetivo de cada inspeção a ser realizada.

A metodologia proposta para a detecção de fraudes também colaborou na compreensão da influência de cada atributo na composição de um perfil de fraude. Através da análise dos conjuntos de atributos, foi possível identificar tanto os atributos imprescindíveis quanto os prejudiciais à detecção dos comportamentos fraudulentos.

O banco de dados utilizado neste trabalho apresentou muitas impurezas e anomalias, demandando um esforço exagerado na etapa de pré-tratamento e consolidação dos dados. Após o tratamento, somente 4,6% dos registros disponíveis para mineração apresentaram resultado de inspeção fraudulento.

Apesar dos pontos negativos acerca da qualidade dos dados disponíveis, alcançou-se taxas de acerto variando de 15% à 40%, conforme o conjunto de regras escolhido para o teste (futuro processo de inspeção). Tais taxas de acerto são superiores àquelas praticadas pela empresa de distribuição de energia elétrica, que variam de 5 à 10%.

7.2 Contribuições

Embora este trabalho tenha abordado especificamente a detecção de fraudes em consumidores de energia elétrica, a metodologia proposta pode ser extendida para a detecção de outros tipos de fraudes, principalmente aquelas em que a ocorrência de fraudadores é menor que 10%. Portanto, este trabalho representa uma importante contribuição, visto que as publicações na área de detecção de fraudes não detalham suas metodologias e resultados, prejudicando o aperfeiçoamento das técnicas e ferramentas contra fraudes.

Este trabalho enunciou em detalhes a fundamentação teoria de Rough Sets, como também apresentou uma abordagem prática da aplicação de seus conceitos. Por este motivo, o trabalho contribui como uma referência ou fonte de estudo da teoria de Rough Sets.

7.3 Trabalhos Futuros

Por ser um trabalho pioneiro na utilização de Rough Sets para detecção de fraudes em consumidores de energia elétrica, alguns pontos merecem um estudo mais aprofundado:

- **Teste Prático da Metodologia**: Pretende-se validar alguns dos conjuntos de regras em processos de inspeção das empresas de distribuição de energia elétrica, de modo a testar na prática a eficiência da metodologia para detecção de fraudes;
- **Discretização de Atributos**: Realizar uma revisão bibliográfica abrangente, principalmente das técnicas estatísticas para este propósito, como a função densidade de probabilidade. Propor otimizações aos algoritmos e heurísticas de discretização apresentados neste trabalho, principalmente em relação à quantidade de memória necessária para esta operação;
- **Reduto**: Aplicar heurísticas de busca por redutos em sistemas de informação, comparando seus resultados com as avaliações de conjuntos de atributos da metodologia;
- ***Toolbox* de Rough Sets**: organizar as implementações deste trabalho em um *Toolbox* de Rough Sets para o MATLAB, permitindo que esta emergente técnica seja aplicada com facilidade em diferentes aplicações.

Referências Bibliográficas

ALESKEROV, E.; FREISLEBEN, B.; RAO, B. Cardwatch: a neural network based database mining system for credit card fraud detection. In: *Computational Intelligence for Financial Engineering (CIFEr), 1997, Proceedings of the IEEE/IAFE 1997*, p. 220–226, 1997.

BOLTON, R. J.; HAND, D. J. Unsupervised profiling methods for fraud detection. In: *Proceedings of the 7th Credit Scoring and Credit Control*, 2001.

BREIMAN, L.; FRIEDMAN, J. H.; OHLSEN, R. A.; STONE, C. J. Classification and regression trees. *Chapman & Hall/CRC*, 1993.

CABRAL, J. E.; PINTO, J. O. P.; GONTIJO, E. M.; REIS., J. Fraud detection in electrical energy consumers using rough sets. In: *2004 IEEE International Conference on Systems, Man, and Cybernetics.*, p. 3625–3629, 2004.

CARUANA, R. A.; FREITAG, D. How useful is relevance? Working Notes of the AAAI Fall Symposium on Relevance, 1994.

CHMIELEWSKI, M. R.; GRZYMALA-BUSSE, J. W. Discretization. *Proceedings of the Third International Workshop on Rough Sets Soft Computing (RSSC'94)*, p. 294–301, 1994.

DORRONSORO, J. R.; GINEL, F.; SÁNCHEZ, C.; CRUZ, C. S. Neural fraud detection in credit card operations. *IEEE Transactions On Neural Networks*, v. 8, n. 4, p. 827–834, 1997.

DOUGHERTY, J.; KOHAVI, R.; SAHAMI, M. Supervised unsupervised discretization of continuos features. *Proceedings of the Twelfth International Conference on Machine Learning*, p. 194–202, 1995.

DUBOIS, D.; PRADE, H. Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems*, v. 17, p. 191–209, 1990.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI Magazine*, v. 17, p. 37–54, 1996.

FAYYAD, U. M.; IRANI, K. B. The attribute election problem in decision tree generation. *Proceedings of the AAAI'92*, p. 104–110, 1992.

GOLDBERG, D. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Professional, 1989.

HAYKIN, S. *Neural networks - a comprehensive foundation*. Prentice Hall, 1998.

HENRIQUES, H. O.; FALCÃO, D. M.; BORGES, C. L. T.; TARANTO, G. N.; MANZONI, A.; ANDRADE, W. S.; VIVEROS, E. C. Aplicações de sistemas inteligentes e processamento distribuído na previsão, localização e minimização de perdas em redes de distribuição, resumo do trabalho de pesquisa e desenvolvimento conjunto da Light S.E.S.A. e da COPPE/UFRJ, 2001.

HU, K.; LU, Y.; SHI, C. Feature ranking in rough sets. *AI Communications*, v. 16, n. 1, p. 41–50, 2003.

HUNG, E.; CHEUNG, D. Parallel algorithm for mining outliers in large database. 1999. Disponível em `citeseer.ist.psu.edu/hung99parallel.html`

INMON, W. H. What is a data warehouse?, prism Tech Topic, 1995.

JOHNSON, D. S. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, v. 9, p. 256–278, 1974.

KIRA, K.; RENDELL, L. A practical approach to feature selection. In: SLEEMAN, D.; EDWARDS, P., eds. *International Conference on Machine Learning*, Aberdeen: Morgan Kaufmann, p. 368–377, 1992.

KOHONEN, T. *Self-organizing maps*. Springer Series in Information Sciences, 1995.

KOMOROWSKI, J.; PAWLAK, Z.; POLKOWSKI, L.; SKOWRON, A. *Rough sets: A tutorial. rough-fuzzy hybridization: A new trend in decision making*. Springer-Verlag New York, 1999.

KOU, Y.; LU, C.; SIRWONGWATTANA, S.; HUANG, Y. Survey of fraud detection techniques. In: *Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control*, p. 749–754, 2004.

KPMG *A fraude no brasil - relatório da pesquisa 2004*. Relatório Técnico, KPMG do Brasil, 2004.

KWON, T. M.; FEROZ, E. H. A multilayered perceptron approach to prediction of the secs investigation targets. *IEEE Transactions on Neural Networks*, v. 7, p. 1286–1290, 1996.

LENARCIK, A.; PIASTA, Z. *Discretization of condition attributes space. in intelligent decision support - handbook of applications and advances of the rough sets theory* Kluwer Academic Publishers, p. 373–389, 1992.

LENARCIK, A.; PIASTA, Z. Probabilistic approach to decision algorithm generation in the case of continuous condition attributes. *Foundations of Computing and Decision Sciences*, v. 18, n. 3–4, p. 213–223, 1993.

LENARCIK, A.; PIASTA, Z. *Probabilistic rough classifiers with mixture of discrete and continuous attributes. in rough sets and data mining - analysis of imprecise data* Kluwer Academic Publishers, p. 373–383, 1997.

MANILLA, H. Finding interesting rules from large sets of discovered association rules. In: *3rd International Conference on Information and Knowledge Management*, 1994.

MINSKY, M. *Society of mind* Simon and Schuster, 1985.

MITRA, S.; PAL, S. K.; MITRA, P. Data mining in soft computing framework: A survey. *IEEE Transactions On Neural Networks*, v. 13, n. 1, p. 3–14, 2002.

MONARD, M. C.; ALVES, G. E.; KAWAMOTO, S.; PUGLIESI, J. B. Uma introdução ao aprendizado simbólico de máquina por exemplos, notas Didáticas do ICMC-USP - São Paulo/SP - Brasil, 1997.

MURTHY, S.; KASIF, S.; SALTZBERG, S.; BEIGEL, R. Randomized induction of oblique decision trees. *Proceedings of the Eleventh National Conference on AI*, p. 322–327, 1993.

NG, A. Y. Preventing "overfitting" of crossvalidation data. In: KAUFMANN, M., ed. *Proceedings of the Fourteenth International Conference on Machine Learning*, Nashville, TN, p. 245–253, 1997.

NGUYEN, H. S. *Discretization of real value attributes, boolean reasoning approach*. Tese de Doutoramento, Warsaw University, 1997.

NGUYEN, H. S.; SKOWRON, A. Quantization of real values attributes: Rough set and boolean reasoning approaches. *Proceedings of the International Workshop on Rough Sets Sof Computing at Second Annual Joint Conference on Information SCiences (JCIS'95)*, p. 34–37, 1995.

NOONAN, J. *Data mining strategies*. Relatório técnico, DM Review, 2000.

PASSINI, S. R. R. *Mineração de dados para detecção de fraudes em ligações de água.* Dissertação de mestrado, Pontifícia Universidade Católica de Campinas - PUC Campinas, 2002.

PAWLAK, Z. Rough sets. *International Journal of Computer and Infomration Sciences*, v. 11, p. 341–356, 1982.

PAWLAK, Z. *Rough sets: Theoretical aspects of reasoning about data*. Kluwer Academic Publishers, 1991.

PAWLAK, Z.; GRZYMALA-BUSSE, J.; SLOWINSKI, R.; ZIARKO, W. Rough sets. *Comunications of the ACM*, v. 38, n. 11, p. 89–95, 1995.

PIATETSKY-SHAPIRO, G. Knowledge discovery in real databases: A report on the ijcai-89 workshop. *AI Magazine*, v. 11, n. 5, p. 68–70, 1991.

PILA, A. D. *Seleção de atributos relevantes para aprendizado de máquina utilizando a abordagem de rough sets*. Dissertação de mestrado, ICMC/USP - São Carlos, 2001.

POLKOWSKI, L.; KACPRZYK, J.; SKOWRON, A. *Rough sets in knowledge discovery 2: Applications, case studies, and software systems*. Physica-Verlag, 1998.

QUINLAN, J. R. Simplifying decision trees. *Int. J. Man-Mach. Stud.*, v. 27, n. 3, p. 221–234, 1987.

QUINLAN, J. R. Induction of decision trees. In: SHAVLIK, J. W.; DIETTERICH, T. G., eds. *Readings in Machine Learning*, Morgan Kaufmann, originalmente publicado em *Machine Learning* 1:81–106, 1986, 1990.

REIS, J.; GONTIJO, E. M.; MAZINA, E.; CABRAL, J. E.; PINTO, J. O. P. Fraud identification in electricity company customers using decision tree. In: *2004 IEEE International Conference on Systems, Man, and Cybernetics*, p. 3730–3734, 2004.

RUSSEL, S. J.; NORVIG, P. *Artificial intelligence: A modern approach*. Prentice Hall, 1995.

SAGE, A. P. *Concise encyclopedia of information processing in systems and organizations*. Pergamon, 1990.

TURING, A. M. Computing machinary and intelligence. *Oxford University Press - Journal of the Mind Association*, v. 59, p. 433–460, 1950.

ZADEH, L. A. Fuzzy logic, neural networks and soft computing. *Comunications of the ACM*, v. 37, p. 77–84, 1994.

ZIARKO, W.; SHAN, N. Kdd-r: A comprehensive system for knowledge discovery in databases using rough sets. In: *RSSC'94 The Third International Workshop on Rough Sets and Soft Computing*, p. 164–173, 1994.