VERA LÚCIA MARCHIORI FALQUETE

UTILIZAÇÃO DE LÓGICA PARACONSISTENTE PARA TRATAMENTO DE INCONSISTÊNCIAS EM SISTEMAS DE RACIOCÍNIO BASEADO EM CASOS

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Curitiba
2004

VERA LÚCIA MARCHIORI FALQUETE

UTILIZAÇÃO DE LÓGICA PARACONSISTENTE PARA TRATAMENTO DE INCONSISTÊNCIAS EM SISTEMAS DE RACIOCÍNIO BASEADO EM CASOS

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: *Sistemas Inteligentes*

Orientador: Prof. Dr. Celso A. A. Kaestner
Co-orientador: Prof. Dr. Júlio César Nievola

Curitiba
2004

Falquete, Vera Lúcia Marchiori

Utilização de Lógica Paraconsistente para Tratamento de Inconsistências em Sistemas de Raciocínio Baseado em Casos. Curitiba, 2004. 108p.

Dissertação (Mestrado) - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

1. RBC (Raciocínio Baseado em Casos). 2. Inconsistência. 3. Lógica Paraconsistente. 4. Fatores Evidenciais. I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática Aplicada.

TERMO DE APROVAÇÃO

Agradecimentos

"Um excelente educador não é um ser humano perfeito, mas alguém que tem serenidade para se esvaziar e sensibilidade para aprender." - Augusto Cury

Em primeiro lugar, a minha gratidão a Deus, por estar sempre presente, por ter me guiado e concedido discernimento para fazer escolhas tão certas quanto as que venho fazendo. E por estar convicta de que a porta que o Senhor abre ninguém fecha.

Ao Profs. Celso Antonio Alves Kaestner e Júlio César Nievola, por repartirem comigo seus conhecimentos, colocando em minhas mãos ferramentas com as quais abrirei novos horizontes, rumo a satisfação plena de meus ideais profissionais e humanos. Em especial ao Prof. Celso pela paciência, orientação e longas discussões sobre os rumos deste trabalho.

Ao Prof. Décio Krause por aceitar participar da banca examinadora e por suas valiosas considerações.

Ao Prof. Bráulio Coelho Ávila, por todo apoio e orientação sobre o caminho a seguir diante de tantas possibilidades.

Aos Profs. Fabrício Enembreck, Sérgio Aparecido Ignácio, Alex A. Freitas e à Profª. Cinthia Obladen de Almeida Freitas, pelas referências de materiais de pesquisas anteriormente realizadas.

À minha família, em especial à minha mãe Odette e ao Marco Antonio, meu esposo, por todo apoio, paciência, carinho e principalmente compreensão. Extensiva também a minha tia Tereza e a minha prima Míriam.

Aos meus amigos(as) Ana Carolina, Cláudia, Dirce, Emerson, Fabiano, Fernanda, Gabriel, Igor, Jaime, João Andrei, Josimeire, Marcelo, Pilar, Rafael, Renata, Simone e Tatiana por todas as palavras de incentivo rumo ao sucesso.

Sou grata a todos que, direta ou indiretamente, contribuiram para que este trabalho fosse realizado.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ABBB.....Árvore Binária de Busca Balanceada

AM.....Aprendizado de Máquina

FNC.....Forma Normal Conjuntiva

IA.....Inteligência Artificial

LP.....Lógica Paraconsistente

MC.....Fator de Crença

MD.....Fator de Descrença

NP.....Não Polinomial

LPA2v.....Lógica Paraconsistente Anotada 2 valores

PrLE.....Programação Lógica Evidencial

RBC.....Raciocínio Baseado em Casos

SAT.....Satisfatibilidade

SE.....Sistemas Especialistas

TP.....Teoria da Probabilidade

RESUMO

Uma das questões centrais de pesquisa em IA (Inteligência Artificial) é a Representação e Manipulação do Conhecimento. Nesta área busca-se pela criação de metodologias que representem mais fielmente aspectos da cognição humana. A aplicação de sistemas RBC (Raciocínio Baseado em Casos) tem se mostrado bastante efetiva em diferentes ramos do conhecimento. O RBC é baseado na idéia de que novos problemas freqüentemente podem ser resolvidos usando soluções passadas. O método básico usado para implementar RBC é construir uma base de casos de problemas previamente resolvidos. Estes casos são então recuperados e adaptados para resolver novos problemas. A partir deste processo, um sistema baseado em casos pode aprender a aperfeiçoar sua capacidade de resolução. Porém, a possibilidade da existência de inconsistências na base de casos é um fator relevante que, embora observado, não tem sido considerado com profundidade nas abordagens apresentadas até agora. Estas inconsistências surgem na forma de contradições entre os casos da base, sejam elas quando um caso se contradiz internamente, seja quando um conjunto de casos provoca uma contradição entre si. Dessa forma, a existência de inconsistências degrada tanto a capacidade de inferência do sistema quanto sua robustez, e reduz a confiabilidade de suas respostas. Torna-se necessário o uso de um formalismo que seja capaz de tratar essas inconsistências, de modo a tornar viável a utilização de RBC em aplicações cotidianas. A LP (Lógica Paraconsistente) é justamente uma das ferramentas mais poderosas para este fim, e pode ser introduzida para evitar o comprometimento da eficácia do sistema. Esse trabalho trata justamente dessa introdução, e da apresentação de algoritmos e formalismos necessários para viabilizar esse processo.

Palavras-Chave: RBC (Raciocínio Baseado em Casos), inconsistências, lógica paraconsistente, fatores evidenciais.

ABSTRACT

One of the central matters of research in AI (Artificial Intelligence) is the Representation and Manipulation of Knowledge. In this area the quest is for the creation of methodologies that more faithfully represent aspects of the human cognition. The application of CBR (Case-Based Reasoning) systems has shown to be very effective in different branches of knowledge. CBR is based on the idea that new problems can often be solved using past solutions. The basic method used to implement CBR is to build a base of cases of previously solved problems. These cases are then recovered and adapted to solve new problems. From this process on, a case-based system can learn how to improve its resolution capacity. However, the probability of existing inconsistencies in the base of cases is an important factor that, although having been observed, has not been considered with depth in the approaches presented until now. These inconsistencies arise in the form of contradictions among cases of the base, be it when a case contradicts itself, when confronted with other cases, or when a set of cases causes a contradiction among them. Thus, the existence of inconsistencies degrades the inference capacity of the system, its robustness, and reduces the reliability of its answers. It becomes necessary the use of a logical formalism that be able to treat these inconsistencies, in order to make possible the use of CBR in everyday applications. PL (Paraconsistent Logic) is just one of the most powerful tools for this end, and can be introduced to avoid compromising the effectiveness of the system. This work deals just with this introduction, as well as with the presentation of algorithms and formalisms needed to make this process viable.

Key-words: CBR (Case-Based Reasoning), inconsistencies, paraconsistent logic, evidential factors.

Capítulo 1

Introdução

1.1 Motivação

"Tudo é duplo; tudo tem dois pólos; tudo tem seu par de opostos; o semelhante e o dessemelhante são uma só coisa; os opostos são idênticos em natureza, mas diferentes em grau; os extremos se tocam; todas as verdades são meias-verdades; todos os paradoxos podem ser reconciliados." - O Caibalion

Raciocinar sobre informações inconsistentes é uma área da Ciência da Computação e IA (Inteligência Artificial) que tem crescido vertiginosamente nos últimos anos. Por outro lado, a área de estudo de Raciocínio Baseado em Conhecimento apresenta uma forma bastante flexível e poderosa para lidar com inferências sobre casos.

Em vista dessas áreas tão promissoras, torna-se um trabalho interessante aproveitar seus pontos fortes e construir um sistema RBC (Raciocínio Baseado em Casos) capaz de não apenas raciocinar sobre um conjunto de casos que representa a experiência do sistema, mas também de lidar com as possíveis inconsistências do mesmo.

A idéia central do RBC consiste em fazer o sistema relembrar de casos relevantes e reutilizá-los em uma nova situação.

Um sistema baseado em conhecimento é composto de programas sofisticados que manipulam a base de conhecimento, implicitamente representada e, usando procedimentos de inferência, heurística e incerteza, tem a capacidade de oferecer ao inquiridor conselhos

inteligentes ou decidir inteligentemente sobre o processamento de uma função e também justificar sua própria linha de raciocínio[1] de maneira direta quando inquiridos. Os problemas resolvidos por esses sistemas são delimitados em uma área específica do conhecimento, e necessariamente são problemas que podem ser simbolicamente representados.

Os sistemas de computação desenvolvidos, em particular na área de IA, precisam e utilizam lógica para seu desenvolvimento. Uma das lógicas mais utilizadas é a Lógica Proposicional Clássica na qual as proposições são tratadas aceitando apenas dois valores: Verdadeiro ou Falso. No entanto, em muitos problemas práticos esta assertiva é de difícil determinação. Daí a importância de utilizar a LP (Lógica Paraconsistente) ao invés da Lógica Clássica.

Segundo Krause [42], dito de modo não muito rigoroso, uma lógica é paraconsistente se pode fundamentar sistemas dedutivos inconsistentes (ou seja, que admitam teses contraditórias, e em particular uma contradição) mas que não sejam triviais, no sentido de que nem todas as fórmulas (expressões bem formadas de sua linguagem) sejam teoremas do sistema.

Em Racine *et al*[62], a estruturação do RBC é amplamente tratada, e como um dos subprodutos são delineadas algumas sugestões para o tratamento de diversos problemas inerentes a uma base de casos, tais como redundância, consistência e inconsistência intra e inter casos. Em particular o problema da inconsistência é uma questão essencial e que, quando tratada, oferece um grande aperfeiçoamento na robustez e confiabilidade do sistema. Por não ser o foco do artigo, Racine *et al*[62], não oferece uma solução de fato. A este trabalho cabe justamente propor uma solução para este problema.

[1]Raciocínio é a atividade criativa que transforma sinais de entrada e conhecimento prévio em novos conhecimentos.

1.2 Objetivos

O presente trabalho tem como objetivo desenvolver algoritmos para encontrar, classificar e tratar inconsistências na base de casos de um RBC, estabelecendo um formalismo capaz de resultar em um conjunto de casos acompanhados de seus respectivos fatores evidenciais, dessa forma utilizando os conceitos de um subcaso da LP, a PrLE (Programação Lógica Evidencial) aplicados ao paradigma RBC. O sistema resultante é capaz de analisar um conjunto de exemplos e gerar dados que possam ser usados pelo RBC para analisar novos casos.

Esse trabalho também procura demonstrar como a LP pode ser usada para tratar e classificar inconsistências e melhorar a performance de classificação de um sistema RBC tradicional.

1.2.1 Problemática

Detectar inconsistências em uma base de casos não é uma tarefa fácil, principalmente quando não dispõe-se de informações externas sobre o que é correto e o que não é. Neste trabalho, o objetivo foi a criação de um sistema que não dispusesse de informações externas, mas ainda assim pudesse melhorar a performance de classificação usando a informação de inconsistência dos casos na base de treinamento.

Naturalmente, quando não há informação externa, existem algumas restrições. Como todo sistema RBC, não é possível uma análise mais simbólica que explicite o raciocínio usado pelo sistema para atribuição de um determinado fator. Dessa forma, a abordagem escolhida é capaz de determinar fatores de credibilidade para cada caso, mas não é capaz de explicar o porquê dos valores atribuídos a esses fatores.

1.3 Contribuição

Até o momento, os sistemas RBC propostos não se utilizaram das técnicas relacionadas à LP para o tratamento de possíveis inconsistências nos casos presentes na base.

Este trabalho vem justamente apresentar uma abordagem inovadora para incorporação dos fatores de crença e descrença da LP para o tratamento de inconsistências em sistemas de RBC. A abordagem desenvolvida utiliza os fatores de crença e descrença como forma de aperfeiçoar a qualidade da classificação. Foi utilizado um classificador K-Vizinhos e incorporou-se esses fatores evidenciais para influenciar o veredito[2]. Levou-se em consideração duas formas: a forma pela qual a distância foi calculada e o cálculo do peso do voto.

1.3.1 Organização do Trabalho

Este trabalho tem a seguinte organização:

Este capítulo apresentou a motivação do trabalho, derivada de sugestões delineadas pelo artigo de Racine *et al*[62], os objetivos do trabalho, a problemática da detecção e tratamento de inconsistências sem dispor de informações externas, e a contribuição almejada pelo trabalho.

O segundo capítulo apresenta a fundamentação teórica do trabalho, definindo um histórico, os elementos de RBC e LP utilizados no trabalho, e alguns dos trabalhos similares já realizados.

O terceiro capítulo apresenta o RBC com Uso de Fatores Evidenciais como escopo propriamente dito do trabalho, isto é, do sistema de detecção e tratamento de inconsistências proposto.

O quarto capítulo apresenta um cenário de experimentação que tem como objetivo ilustrar as técnicas desenvolvidas neste trabalho. Contém um ambiente de testes realizados com a Base Candidato (criada como exemplo) e a descrição de suas características, apresentando sobre a mesma os passos para o cálculo dos fatores evidenciais, da distância e da aplicação do classificador K-Vizinhos. Também contém as principais descrições das

[2]Dado um conjunto de casos, um classificador K-Vizinhos deve achar qual a classe a ser atribuída, baseado nesse conjunto. Essa decisão é denominada veredito, e cada caso influencia essa decisão *votando* em sua própria classificação.

características das outras nove bases utilizadas para os testes.

O quinto capítulo apresenta os experimentos e resultados validando as técnicas desenvolvidas e compara diferentes formas de considerar os casos determinados pelo K-Vizinhos para a classificação. Mostra testes realizados em mais nove bases originadas do UCI [11] (Repository of Machine Learning Databases), da Universidade da Califórnia, os resultados para cada fórmula de votação e os testes fazendo uma comparação dos resultados obtidos em percentuais de precisão e recobrimento.

O sexto e último capítulo apresenta as considerações finais e também extensões e trabalhos que futuramente poderão ser desenvolvidos.

Capítulo 2

Fundamentação Teórica

2.1 Raciocínio Baseado em Casos

2.1.1 Histórico e Estado da Arte

Em meados da década de 70, pesquisadores como Schank [66] e Minsky [52], demostraram interesse na compreensão do raciocínio humano. Estes trabalhos iniciais evoluiram para teorias independentes de aprendizado e cognição[1] [67], [53] como exemplos. Tais teorias possibilitaram a evolução de um extenso campo de pesquisas denominado Ciência Cognitiva[2] e de uma sub-área de IA: RBC [39].

Schank *et al*[69] propuseram que o conhecimento humano podia ser armazenado na forma de um conjunto de *scripts*[3]. Este trabalho é considerado por muitos pesquisadores como sendo uma das principais origens do RBC [1, 81]. Entretanto, Aamodt *et al*[1] consideram que o trabalho de Ludwig Wittgenstein, em 1953, pode ter sido a base filosófica para o RBC.

Schank [67] apresentou seus estudos sobre memória dinâmica e como a manipulação de casos passados e padrões de situação poderiam ser aplicados à resolução de problemas e ao aprendizado. O padrão que Schank definiu agrupa um conjunto de casos com ca-

[1]Cognição compreende a aquisição de conhecimento, capacidade de reconhecimento do que se aprende, utilização do aprendizado, memória, inteligência, linguagem e razão.

[2]Ciência Cognitiva é uma ciência interdisciplinar que abrange: lingüística, filosofia, biologia, neurociências, psicologia e computação. Este campo de estudo não se preocupa apenas em como estudar a percepção do conhecimento humano, mas sim em simulá-lo espontaneamente.

[3]*Script* é uma estrutura que descreve uma sequência estereotipada de eventos em um contexto particular [69]

racterísticas similares, no qual os casos são caracterizados pelos episódios aos quais estão associados.

Segundo Watson [81], o modelo teórico para analogia proposto por Gentner [28] também foi de grande relevância para a área de RBC. Janet Kolodner [35], também foi uma das precursoras em RBC. Um dos primeiros sistemas que utilizou esta abordagem foi o CYRUS. Este sistema, baseado em um modelo proposto por Schank, possuía um modelo de memória onde casos específicos com propriedades similares eram organizados em estruturas mais generalizadas.

Segundo Aamodt *et al*[1], o modelo de memória de casos utilizado para o desenvolvimento do CYRUS foi a base de muitos outros sistemas RBC, incluindo principalmente os seguintes: MEDIATOR [72], PERSUADER [75], CHEF [30], JULIA [32].

Outro trabalho considerado muito relevante para o RBC foi desenvolvido por Porter [1, 81]. Porter *et al*[60], aplicou a abordagem do aprendizado por conceitos para classificação de tarefas. Este trabalho serviu como base para o desenvolvimento do sistema PROTOS. Este sistema enfatizava a integração do conhecimento geral sobre o domínio com o conhecimento de casos específicos dentro de uma estrutura de representação de um campo não definido. Depois, o sistema GREBE, uma aplicação no domínio da lei combinou casos com conhecimento de domínios gerais. O HYPO, sistema desenvolvido para interpretar uma situação na corte e produzir argumentos para ambas as partes, foi uma outra contribuição significativa para RBC. Para otimizar a performance em sistemas baseados em conhecimento, o CASEY foi criado por Kotton [41] do MIT.

Desde a década de 90, RBC tem sido um campo de grande interesse. RBC é freqüentemente utilizado como um termo genérico para descrever técnicas que utilizam raciocínio por analogia.

2.1.2 Funcionamento do Raciocínio Baseado em Casos

Pode-se entender o paradigma RBC como a solução de novos problemas por meio da utilização de casos anteriores já conhecidos [80]. Segundo Schank [68], RBC significa raciocínio sobre exemplos prévios. Desta forma, deve-se inicialmente determinar as similaridades entre o problema a ser resolvido e os casos armazenados na memória e, uma vez determinado o caso mais semelhante, adaptar a solução deste caso para que se possa

resolver o problema em questão.

Conseqüentemente, problemas futuros têm grandes chances de serem semelhantes aos problemas atuais e a utilização de técnicas de recordação e reutilização de conhecimento compõe uma estratégia bastante efetiva de raciocínio. Em linhas gerais pode-se dizer que RBC reutiliza casos para [38]:

- explicar novas situações;
- encontrar novas demandas;
- interpretar novas situações;
- criticar novas soluções;
- criar uma solução para um novo problema.

Uma importante justificativa que apóia a utilização de RBC é que todo esforço feito em uma determinada situação passada será desperdiçado se não for armazenado. É possível reutilizar inclusive, insucessos de forma a antever uma situação de falha e assim poder evitá-la.

Pode-se, desta forma, considerar que um raciocinador ao reutilizar uma experiência prévia obtenha em geral soluções com maior qualidade, uma vez que possui maior competência para solucionar este tipo de problema [38].

Para sistemas em RBC terem sucesso deverão se preocupar em como os casos serão organizados na memória, como serão recuperados da memória, como casos anteriores serão adaptados a novos problemas e como serão adquiridos.

Entre os principais fatores que influenciam a qualidade de uma solução que reutilize casos, estão:

- a experiência apresentada pelo raciocinador;
- os casos e experiências que possui;
- sua capacidade de interpretar novas situações em termos de experiências prévias;
- sua aptidão para realizar a adaptação;

- sua aptidão para realizar a avaliação.

Cabe ressaltar que os estudos em RBC são incapazes de simular todas as condições do comportamento cognitivo humano, mas devem apresentar as seguintes suposições psicológicas que este paradigma apresenta [73]:

- a memória é predominantemente episódica[4] de fatos que representam as experiências conhecidas pelo sistema.
- a memória armazena experiências e estas conduzem o raciocínio, uma vez que a interpretação e a compreensão de novas situações se dão sobre experiências já conhecidas;
- a memória é extremamente indexada, podendo uma mesma experiência ser representada por índices ou caminhos diferenciados;
- a memória é dinâmica, sendo possível ocorrer mudanças em sua estrutura com o passar do tempo.

Uma ilustração genérica do raciocínio geral esperado pelo RBC pode ser visto na Figura 2.1.1.

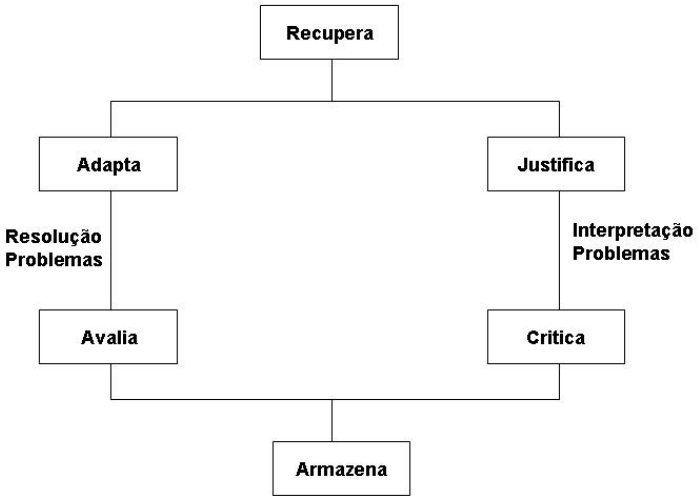
2.1.3 Casos, Fatos e Exemplos

Pode-se considerar um caso como uma abstração de fatos e eventos [6]. Um caso inclui um conjunto de fatos válidos na situação inicial, um conjunto de fatos válidos na situação final e um conjunto de eventos e sua correspondente ordenação [51].

É possível considerar dentro da composição de um caso o contexto e a avaliação da solução [12]. O contexto pode ser usado para resolver ambiguidades, para selecionar um método de resolução de problemas e até mesmo para melhor entender a solução. Já a inclusão de uma avaliação da solução contribui para melhor caracterizar onde a solução se apresenta mais adequada e procura eliminar a recuperação de soluções que aparentemente apresentavam potencialidade de aplicação, mas se mostraram falhas em reutilizações anteriores.

[4]Conhecimento episódico é aquele construído em cima de fatos ocorridos (episódios) e normalmente é pouco estruturado.

Figura 2.1.1: Ilustração Genérica do Processamento em RBC.



Fonte: Livro A Tutorial Introduction to Case-Based Reasoning: Experiences, Lessons, & Future Directions, Kolodner *et al*[40].

Para melhor caracterizar um caso é importante observar os diferentes níveis de abstração a serem considerados [6]:

- de conhecimento: o caso é um resultado comportamental de um processo;
- representacional: caracterização dos componentes de um caso;
- de implementação: especificação da estrutura de representação a ser usada.

Portanto, um caso é extraído no nível de conhecimento, caracterizado no nível representacional e estruturado no nível de implementação.

Um método para escolha do melhor caso é utilizar as heurísticas da preferência [36]:

- orientada por objetivos: dar preferência aos casos que possuam o mesmo objetivo da situação atual;
- por características notáveis: dar preferência aos casos que tenham o maior número de características importantes similares;
- por especificidade: dar preferência aos casos que possuam idênticas características, ao invés dos que possuem características genéricas;

- por frequência: dar preferência a casos que frequentemente são similares a situação atual;
- recentidade: dar preferência a casos que recentemente são similares a situação atual;
- por facilidade de adaptação: dar preferência aos casos com características que sejam facilmente adaptadas a novas situações.

Dependendo do conteúdo do caso ele pode ser usado para diferentes propósitos, tais como [37]:

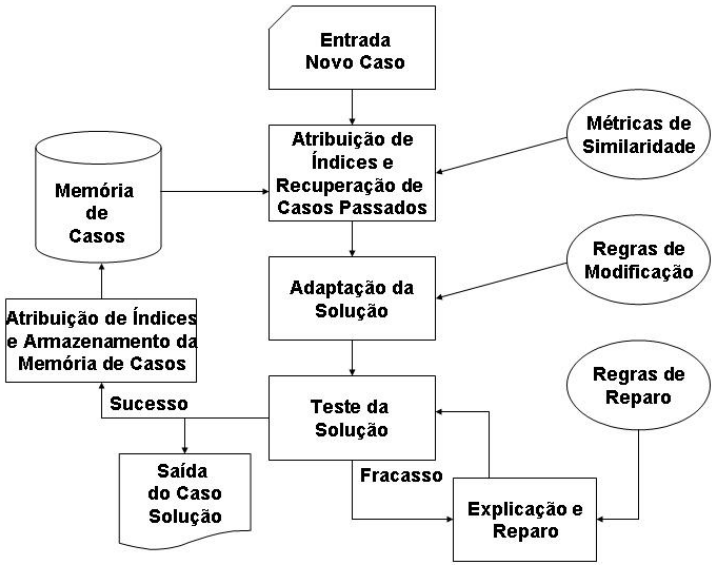
- casos que incluam o problema e a sua solução podem ser usados para derivar ou avaliar soluções para novas situações que possam ocorrer no mesmo domínio de aplicação;
- casos que apresentam a descrição de uma situação e algum resultado podem ser usados para avaliar novas situações;
- casos que descrevem falhas ocorridas podem ser usados para antecipar falhas potenciais em situações futuras;
- casos que contenham a explicação causal das falhas podem ser usados pelos processos de correção de falhas;
- casos que sejam inviáveis para reutilização no problema corrente, mas que contenham uma explicação dos métodos aplicados para derivar sua solução, podem propriciar um indicativo de quais métodos podem ser utilizados.

Os casos apresentam diferentes tipos de relevância funcional em face do tipo de aplicação de RBC que será trabalhado [57], ou seja, a importância do caso depende da funcionalidade da aplicação que trabalhará com o RBC em questão.

2.1.4 Processamento Básico em Raciocínio Baseado em Casos

Na resolução de problemas aplicando RBC, uma solução para um novo caso é obtida recuperando casos similares anteriormente analisados e derivando suas respectivas soluções de modo a se adequar ao novo problema.

Figura 2.1.2: Diagrama do Fluxo de Processamento de Casos do RBC.



Fonte: Arquitetura de um Sistema CBR, Lira *et al*[44].

Na Figura 2.1.2, apresenta-se o processo geral de resolução.

Inicialmente, um novo caso é apresentado ao sistema. Em face do novo problema, utiliza-se um conjunto de métricas de similaridade para determinar quais casos anteriores [6] mais se assemelham ao caso proposto, bem como são determinadas as características chaves utilizadas nessa comparação.

Em seguida, o processo de adaptação consiste em aplicar regras válidas de transformação que procuram alterar as soluções previamente utilizadas para que proponham uma nova solução que satisfaça características consideradas chave no novo problema.

Na etapa de testes, a solução é aplicada sobre o caso de entrada e estima-se o quão bem sucedido foi esse procedimento. O processo de estimativa pode ser dramaticamente diferente para classes de problemas distintas. Normalmente, um profissional da área determina quais os critérios adequados, enquanto o desenvolvedor cria uma função que quantifica o nível de sucesso.

Determinado o grau de sucesso, algum critério é utilizado para decidir se a solução é satisfatória. Em caso de fracasso, o sistema pode tentar determinar os fatores responsáveis e aplicar um conjunto de regras na tentativa de corrigir o problema. Para evitar o risco do sistema entrar em um ciclo interminável (loop), pode-se determinar algum critério de

parada, ao qual o sistema responde que foi incapaz de solucionar o problema.

Em caso de sucesso, além de emitir uma resposta como saída, a nova solução é assimilada pelo sistema, ou seja, armazenada na memória de casos [6].

Na prática, tanto a inclusão como a forma de implementação de cada um dos processos sofrem variações em função de sua estrutura interna [5].

2.2 O Problema Intra-Casos

Embora a abordagem adotada não seja capaz de discernir entre os diferentes tipos de inconsistências, vale a pena conhecer um pouco quais tipos de inconsistências foram estudadas no passado. Uma das classificações possíveis distingue entre inconsistências intra-casos e inter-casos.

Uma inconsistência é denominada intra-casos quando os valores atribuídos para diferentes características dentro de um caso único violam uma ou mais restrições impostas, ou seja, analisando as informações anteriores detecta-se que o caso contém informações contraditórias.

Um exemplo de quando a inconsistência intra-casos acontece é abaixo descrito:

- Seja um caso $c(v_1, ..., v_i, ..., v_j, ..., v_n)$
  - Onde v_i é o valor do i-ésimo atributo
  - v_n é o valor do n-ésimo atributo
  - v_i e v_j violam restrições do domínio

Observe que a detecção de inconsistência intra-casos exige conhecimento do domínio de casos. Dessa forma, ou o domínio é fornecido, ou então o sistema deverá derivar um domínio incrementalmente a partir dos casos fornecidos.

Uma representação comum do domínio é através de regras que o restringem, por exemplo:

> "Se a pessoa fala português, nasceu na América Latina,
> e não se naturalizou em outro país que não a terra natal,
> então deve ser de nacionalidade brasileira."

Uma regra como a descrita acima pode ser facilmente imaginada por um ser humano, mas derivar regras como essa em computador é uma tarefa complexa e que faz algumas exigências quanto aos dados de entrada. Para que o sistema seja capaz de criar regras que restrinjam o domínio corretamente, o mínimo necessário é fornecer um conjunto de casos consistentes como base para aprendizado. Mas o problema que está sendo tratado é justamente o fato da base de casos de entrada possuir inconsistência que são desconhecidas, portanto tornando quase impossível determinar regras explícitas de restrição de domínio.

2.3 O Problema Inter-Casos

Enquanto a inconsistência intra-caso surge ao analisar apenas o caso isoladamente, a inconsistência inter-casos surge quando confrontados dois ou mais casos. Por essa razão, a detecção da inconsistência inter-casos exige comparação de todos os possíveis subconjuntos de casos. De fato, este problema é tão complexo que está dentro da classe de problemas NP-Completos[5]. Uma vez que não foi encontrado na literatura nenhuma prova formal dessa afirmação, esta seção procura demonstrar a NP-Completude do seguinte problema:

> *Dado um conjunto de casos, cada caso descrito por um conjunto de atributos e uma classificação, que é determinada em função dos atributos do caso, verificar se existe contradição (inconsistência) entre os casos.*

Uma vez que a classificação de cada caso está em função de seus atributos, pode-se fazer uma analogia com um problema de lógica, no qual existe um conjunto de variáveis relacionadas em uma expressão lógica e que implicam que uma outra relação é verdadeira, no caso do presente trabalho a variável de classificação assumir um determinado valor. Dessa forma, uma contradição surge quando dois conjuntos de valores dados aos atributos deveriam resultar na mesma classificação quando avaliado pela relação escolhida, mas cujas classificações informadas são diferentes. Vale a pena observar que no contexto desse trabalho esta relação é desconhecida.

Considere uma configuração de atributos de um caso que seja suficiente para levar a alguma conclusão sobre o caso, embora a conclusão não seja necessariamente especificada com antecedência. Essa configuração pode ser vista como um conjunto de atributos valorados. A conclusão pode estar sendo representada por algum dos atributos do caso.

Seja CI um conjunto de casos que é definido de modo que todo o caso c_i possua a configuração mencionada anteriormente. Uma inconsistência é denominada inter-casos quando algum dos casos que deveriam chegar a uma mesma conclusão por força dos valores dos atributos, chegam a conclusões diferentes.

[5]Um problema P é considerado NP-Completo se estiver em NP, que são problemas solucionáveis em tempo polinomial em uma máquina de turing não determinística, e se todos os outros problemas em NP são redutíveis para P. Em termos práticos, isso significa que somente se conhecem algoritmos deterministas de complexidade exponencial ou pior para resolver os problemas NP-Completos.

O problema de detecção de inconsistências inter-casos será considerado através de uma abordagem baseada no cálculo de fatores de crença e descrença[6], uma vez que existe uma forte suspeita de que o problema é de complexidade exponencial[7], como mostra-se adiante.

Para sustentar essa suspeita, pode-se fazer uma analogia com o problema NP-Completo SAT (Satisfatibilidade) [25, 70, 58, 15].

Uma característica essencial de todo NP-Completo é que dados dois problemas P e P' NP-Completos, P pode ser descrito como um problema do tipo P'.

Considere uma fórmula da Lógica Proposicional com n variáveis para a qual deve-se encontrar um conjunto de valores para as variáveis mencionadas tal que a fórmula seja avaliada como verdadeira. A fórmula em questão deve estar representada na FNC (Forma Normal Conjuntiva), ou seja, pode-se ver o problema como um conjunto de disjunções.

Exemplo:

- Dado uma fórmula F da lógica proposicional
- Sejam $v_1, v_2, ..., v_n$ as variáveis booleanas
- O problema é determinar se F pode ser satisfeito

Uma expressão booleana é satisfatível quando existe um conjunto de valores para suas variáveis que a tornam verdadeira.

Se este problema tem 3 ou mais variáveis, é comprovadamente NP-Completo [25, 58, 15].

Exemplo:

$$B = (x_1 \vee x_2) \wedge (x_3 \vee x_1 \vee x_2) \wedge (x_2 \vee x_3) \tag{2.1}$$

A seguir, é apresentada uma analogia que mostra a semelhança entre os dois problemas.

Considere um conjunto de casos cuja consistência deve ser verificada. Supondo a existência de N casos. Cada caso pode ser representado como:

[6]Fator de Crença e Descrença indica o quanto se acredita na veracidade de uma informação ou não.

[7]Um problema de complexidade exponencial é aquele cujo tempo de execução segue uma função exponencial em relação ao tamanho da entrada.

```
Caso_n{
  a_1 :  V
  a_2 :  F
  ...
  a_i :  ?
  ...
  a_j :  ?
}
```

Onde a_i são os atributos do caso.

Cada atributo do caso pode receber somente valores V ou F. Observe que existem atributos cujo valor é desconhecido. Resumindo o caso a apenas atributos cujo valor é desconhecido:

```
Caso_n{
  a_i :?
  a_{i+1} :  ?
  ...
  a_j :  ?
}
```

Como os atributos acima não tem valores, a princípio pode-se atribuir qualquer valor aos atributos, mas aqui limita-se os valores somente a V (Verdadeiro) ou F (Falso).

Do ponto de vista de apenas um caso, o que se estaria procurando seriam os valores de a_i, a_{i+1}, ..., a_j que satisfaçam as regras de domínio (as quais por sua vez, devem considerar os atributos que já tem valor).

Antes de continuar, faz-se uma consideração sobre as regras de domínio. Qualquer regra de domínio vai estabelecer alguma relação entre as variáveis. As relações podem ser reduzidas para serem visualizadas como expressões booleanas nesse caso, porque as variáveis só podem receber dois valores.

Qualquer restrição vai exigir que um conjunto de variáveis ao mesmo tempo assuma um determinado valor.

Exemplo:

"`lingua=portugues`"

Como os valores devem ser assumidos ao mesmo tempo, isso indica que existe uma relação $\wedge$(E) entre eles.

Exemplo:

"`lingua=portugues` $\wedge$ `local_nascimento=brasil`"

Cada variável só pode assumir dois valores, ou **V** ou **F**.

Dessa forma, pode-se escrever:

"`brasileiro=V` $\wedge$ `fala_portugues=F`"

Finalmente, faça a seguinte consideração:

Atributo a_i = V $\rightarrow$ escreve-se a_i na expressão

Atributo a_j = F $\rightarrow$ escreve-se $\neg$ a_j

Dessa forma, tem-se:

"`brasileiro` $\wedge$ $\neg$`fala_portugues`"

Note que ao final, o exemplo apresenta uma expressão na lógica booleana.

Outra observação importante é que **toda** expressão booleana pode ser escrita na Forma Normal Conjuntiva. A expressão na FNC mais simples é uma disjunção (a $\vee$ b $\vee$ ... $\vee$ z). Ainda, a disjunção é a expressão booleana mais fácil de ser satisfeita, porque basta que uma variável tenha um valor V.

Dessa forma, qualquer problema que tenha uma expressão mais complicada será ainda mais difícil do que no caso da disjunção. Por isso, nesta dedução supõe-se o caso mais simples que é uma disjunção de todas as variáveis (atributos).

Assim, os atributos tem que ser relacionados da seguinte forma:

$$(a_i \vee a_{i+1} \vee ... \vee a_j)$$

Note que todos os casos terão uma expressão como esta. O objetivo aqui é verificar se os casos não se contradizem. Observe que de um caso para outro, deseja-se que **ambos** os casos sejam satisfeitos ao mesmo tempo.

Isso significa que tem-se disjunções conectadas pelo AND, ou seja, uma expressão na FNC.

Então, os casos não se contradizem se for possível atribuir valores para todas as variáveis que satisfaçam todas as restrições, o que equivale a verificar se uma expressão lógica na FNC pode ser satisfeita. Ou seja, o problema é uma instância do problema SAT.

Essa analogia foi feita em cima de uma redução do problema, ou seja, em uma versão simplificada. Isso indica que, no caso geral, o problema de detecção é NP-Completo porque pode ser reduzido para um SAT.

Pode-se questionar como o problema seria tratado no caso de não existirem atributos sem valor. Ora, neste caso não existem variáveis e o problema pode ser simplificado.

Uma possível solução nesta configuração pode ser apenas comparar as conclusões. Dado um determinado conjunto de atributos, se eles forem iguais, a conclusão deveria ser a mesma, do contrário tem-se uma contradição. Observe que bastaria uma comparação dois a dois para detectar inconsistência nesse caso.

Na próxima subseção, será explicado porque a comparação dois a dois não é suficiente para detectar inconsistências no caso geral.

2.3.1 Confrontando Casos para Detectar Inconsistências

Considerando, então, que cada caso pode ser visto como uma expressão da lógica proposicional, o seguinte exemplo mostra porque uma comparação dois a dois, ou mesmo qualquer comparação que limite o número máximo de equações consideradas (confrontadas entre si) ao mesmo tempo, pode não ser capaz de detectar uma inconsistência.

Suponha que o seguinte conjunto de expressões faz parte de uma base de fatos:

- $a \vee b$
- $a \vee \neg b$
- $\neg a \vee b$

Tabela 2.3.1: Espaço de Possibilidades de Valores para A e B.

A	**B**
F	F
F	V
V	F
V	V

- $\neg a \vee \neg b$

Os passos seguintes verificam se esse conjunto de equações é consistente. Assumindo que A e B só podem ter valores V (Verdadeiro) e F (Falso), tem-se o seguinte espaço de possibilidade de valores para A e B conforme Tabela 2.3.1.

A metodologia para solucionar esse problema é a seguinte: para cada equação verifica-se quais valores do espaço de possibilidades **não** podem satisfazer a equação. Esse procedimento é fácil de aplicar nesse caso já que, como todas as cláusulas são compostas apenas de uma disjunção, para cada cláusula existe apenas uma combinação que não satisfaz a equação. Então, na ordem, são apresentadas quais combinações não satisfazem cada cláusula.

1. Para a primeira equação, $A = F$ e $B = F$ não satisfazem.
2. Para a segunda equação, $A = F$ e $B = V$ não satisfazem.
3. Para a terceira equação, $A = V$ e $B = F$ não satisfazem.
4. Para a quarta equação, $A = V$ e $B = V$ não satisfazem.

Ora, mas depois do processo de eliminação, não restou nenhuma solução. Isso mostra que esse conjunto é inconsistente. O fato importante a observar é que qualquer comparação que não tivesse considerado todos os casos ao mesmo tempo não seria capaz de detectar que existe essa inconsistência.

Pode-se observar que para qualquer número de váriaveis, é possível construir um conjunto de equações que elimina cada uma das soluções possíveis e que, desta forma, só seria possível detectar que estas equações são inconsistentes considerando todas ao mesmo tempo. Por esta observação, fica claro que qualquer solução que limita o número de

equações confrontadas não é capaz de garantir que não existe um conjunto maior de equações que seja inconsistente.

2.4 Representação Lógica

A lógica foi originalmente desenvolvida para formalizar os princípios de raciocínio válido. Isto tem sido estudado desde os tempos de Aristóteles, embora a chamada lógica moderna tenha início em 1879, data em que Gottlob Frege publicou a primeira versão do que hoje é conhecido como cálculo de predicados [8].

A Lógica procura fazer do raciocínio envolvido em Ciência ou Matemática um processo rigoroso, sendo naturalmente utilizada em áreas onde a prova dedutiva é requerida. Como exemplo, a prova de que um programa de computador produz o resultado expresso na sua especificação. Entretanto, o problema de representação do conhecimento, refere-se não a domínios formais, mas a problemas do cotidiano, que são resolvidos por raciocínio informal e que, muitas vezes, são de difícil caracterização.

2.4.1 Lógica Proposicional

A Lógica Proposicional tenta abstrair as características essenciais do raciocínio dedutivo e expressá-los no que pode ser chamado de uma álgebra de proposições[8]. Por exemplo:

- Raiz quadrada de dois é um número irracional.
- O albatroz é um mamífero E vive próximo do mar.

Observe que o exemplo acima descrito contém duas proposições, considerando a interpretação usual, a primeira é (V) Verdadeira e a segunda é (F) Falsa, porque embora albatrozes vivam próximos do mar, não são mamíferos.

A Lógica Proposicional é definida em dois níveis bem distintos. No primeiro nível ela é uma linguagem formal com regras de formação para gerar sentenças - fórmulas bem formadas da linguagem. Neste nível as proposições são comumente denotadas por símbolos, tais como: p, q, r, s dentre outros. Uma correspondência pode ser feita entre os símbolos da linguagem e objetos ou valores em algum domínio. Essa correspondência, ou mapeamento para um domínio, é conhecida como interpretação, e corresponde ao segundo nível [78].

[8]Proposições são sentenças que podem ser consideradas como: F (Falso) ou V (Verdadeiro) não sendo permitidos outros valores.

Tabela 2.4.2: Conectivos Utilizados na Lógica Proposicional.

Símbolo	**Significado**	**Denominação**
$\wedge$	e	conjunção
$\vee$	ou	disjunção
$\neg$	não	negação
$\rightarrow$	implica	implicação

Tabela 2.4.3: Utilização de Símbolos e Conectivos Produzindo Sentença com Valores (v,f).

Sentença	**Valor**
p$\wedge$q	é **v** somente se ambos p e q são **v**
p$\vee$q	é **v** se pelo menos um p e q é **v**
$\neg$p	é **v** se p é **f** e vice-versa
p$\rightarrow$q	é **v** a não ser que p é **v** e q é **f**

Expressões mais complexas na linguagem são construídas utilizando um conjunto de símbolos, conhecidos como conectivos. A Tabela 2.4.2 apresenta os conectivos tipicamente mais utilizados em Lógica Proposicional.

A utilização de símbolos e conectivos produz sentenças, com os seguintes valores (v,f) que são apresentados na Tabela 2.4.3.

2.4.2 Lógica Paraconsistente

Raciocinar sobre informações inconsistentes é uma área da Ciência da Computação e da IA que tem crescido vertiginosamente nos últimos anos. Como visto anteriormente, a Lógica Proposicional Clássica trata proposições segundo o seu valor lógico V (Verdadeiro) ou F (Falso). No entanto, em muitos problemas práticos esta assertiva é de difícil determinação.

Por exemplo, considera-se o caso [79] de um diagnóstico de um determinado médico sobre a ocorrência de uma doença no paciente P. Suponha-se que o médico M1 conclua que o paciente P contraiu a doença D. Na hipótese do paciente P procurar um outro médico M2, e este diagnosticar que o paciente "não"contraiu a doença D, qual a conclusão que o paciente chegaria? Uma das opções seria procurar um terceiro médico M3 para tirar a dúvida. Enriquecendo ainda mais a questão, suponha-se que os médicos apresentem seu diagnóstico mas não com 100% de certeza, tanto o médico M1 quanto o médico

M2 tem dúvidas quanto à presença ou não da doença. A LPA2v (Lógica Paraconsistente Anotada com 2 valores), um subcaso da LP, é um paradigma útil para tratar das situações intermediárias entre Verdadeiro e Falso. Com a notação dos Fatores de Crença e de Descrença numa informação, tem-se a opção do especialista explicitar o quanto acredita e desacredita na informação.

Exemplo:

- Paciente P tem a Doença D?
- Médico M1 [0,8; 0,4] Significa dizer que o médico M1 tem 80% de crença e 40% de descrença.
- Médico M2 [0,2; 0,9] Significa que o médico M2 tem 20% de crença e 90% de descrença na incidência da doença.

Observa-se que os valores de crença e descrença não são índices complementares cuja soma tem de ser um, ou seja, os valores são considerados pelo especialista de maneira independente.

O valor Verdadeiro aparece na forma [1,0; 0,0] e o valor Falso [0,0; 1,0]. Com esta notação, outros valores são de grande importância:

- [1,0; 1,0] representando uma inconsistência na informação;
- [0,0; 0,0] representando uma indeterminação na informação (não existe informação);
- [0,5; 0,5] representando uma indefinição na informação (a informação existe, mas há dúvida sobre qual está correta).

Este exemplo ilustra o fato de que mesmo especialistas de uma mesma área, podem divergir em determinado diagnóstico [79]. Da mesma forma, é normal a existência de informações contraditórias em qualquer área do conhecimento. Foi para tratar estas contradições que surgiu então a LP.

Lógica Paraconsistente e sua Notação

Seja L uma lógica e L' uma linguagem, que se supõe conter o símbolo de negação $\neg$. Uma teoria T, que tem por base L, é um conjunto fechado de sentenças pelas inferências aceitas por L; ou seja, T contém todas as conseqüências (via L) de suas sentenças. As sentenças de T são seus teoremas.

T diz-se inconsistente se existir uma sentença A tal que A e $\neg A$ sejam teoremas de T, em caso contrário T denomina-se consistente. T é trivial se qualquer sentença de L' for teorema; em hipótese contrária, T é denominado não trivial [17].

Uma lógica é dita paraconsistente (LP), se admite teorias inconsistentes mas não triviais.

A LPA2v estende a Lógica Clássica bi-valorada, adicionando duas anotações, desconhecido e inconsistente. Dessa forma, os quatro anotações possíveis são as seguintes:

- $V \rightarrow (1,0)$ ou seja, verdadeiro;
- $f \rightarrow (0,1)$ ou seja, falso;
- $\perp \rightarrow (0,0)$ ou seja, desconhecido;
- $\top \rightarrow (1,1)$ ou seja, inconsistente.

Os valores da LP também podem ser representados através de um reticulado, tal como observado na Figura 2.4.3.

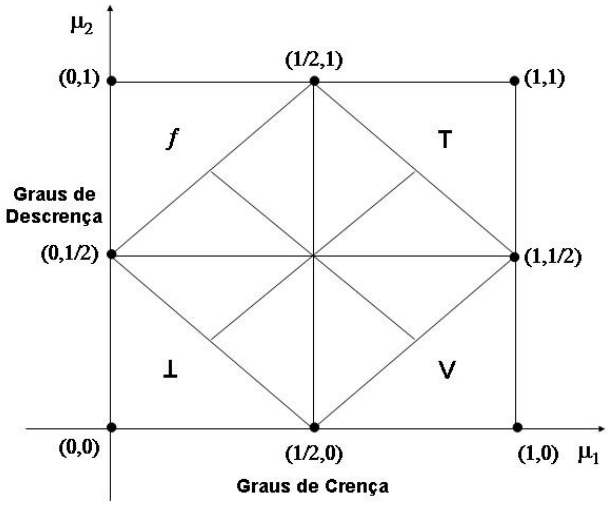
Para ilustrar cada uma das anotações, considere a seguinte proposição:

- p = "Vera possivelmente tem gripe."

Representa-se $p(\mu_1, \mu_2)$, no qual μ_1 é o grau de crença e μ_2 é grau de descrença da proposição p. Dessa forma, pode-se estudar p segundo os diferentes valores de crença e descrença:

- No caso de crença total $p \rightarrow p(1,0)$, acredita-se que a proposição seja 100% verdadeira.
- No caso de descrença total $p \rightarrow p(0,1)$, acredita-se que a proposição seja 100% falsa.

Figura 2.4.3: Espaço de Possibilidades.



Fonte: Livro Lógica Paraconsistente Aplicada, Costa [17].

- No caso de inconsistência total $p \rightarrow p(1,1)$, acredita-se que existem evidências igualmente confiáveis que se contradizem.
- Finalmente, existe o caso indefinido $p \rightarrow p(0,0)$, no qual não existem informações sobre a proposição.

2.4.3 Programação Lógica Evidencial Paraconsistente

Em 1910 foram publicados, de forma independente, os primeiros trabalhos desenvolvidos para tratar sistemas lógicos inconsistentes que foram realizados pelo lógico russo Nicolai A. Vasiliev e pelo lógico polonês Jean Lukasiewicz. Estes trabalhos apresentavam uma lógica onde as contradições não eram eliminadas, mas se restringiam à Lógica Aristotélica Tradicional, no que se refere a paraconsistência.

Somente em 1948 o lógico polonês Stanislaw Jáskowski, e em 1954 o filósofo e matemático brasileiro Newton C. A. da Costa [17], respectivamente, construíram independentemente a LP. Mas foi em 1963 que este brasileiro Newton C. A. da Costa rompeu definitivamente com o aristotelismo ao tomar como objeto de estudo a contradição.

A partir dos anos 70, devido à sua aplicação a LP evoluiu de modo muito rápido.

A introdução dos estudos em PrLE [74, 9, 10], possibilitou a criação de uma extensão da

Linguagem Prolog, o Paralog [18] e Paralog-e [79], permitindo desta maneira o tratamento do fenômeno da inconsistência.

A PrLE associa a cada proposição lógica os fatores evidenciais de crença e descrença às anotações desta proposição. Deste modo uma determinada proposição é anotada da seguinte forma:

$p = (crença, descrença)$

Onde:

- *crença*: fator que indica o quanto se acredita na verdade da proposição, sendo um número real que varia entre 0(nenhuma crença) e 1 (crença absoluta);
- *descrença*: fator que indica o quanto se desacredita da verdade da proposição (ou se acredita na falsidade da proposição), sendo um número real que varia entre 0 (nenhuma descrença) e 1 (descrença absoluta).

Pode-se concluir que a PrLE fornece um modelo de raciocínio que não elimina a presença de informações contraditórias. Existem infinitas anotações possíveis para uma premissa. Essa característica permite quantificar a inconsistência dos itens de conhecimento envolvidos.

2.5 Trabalhos Similares

Este trabalho se deve muito ao trabalho de Racine *et al*[62] e Yang *et al*[63], em particular no tocante à motivação. Os trabalhos sobre RBC tinham como objetivo principal o tratamento de inconsistências propondo soluções para detecção de deficiências no RBC tais como redudância e inconsistência intra e inter casos. As soluções propostas por Racine e Yang apoiavam-se em uma abordagem semi-automatizada para detecção, na qual era necessário que um especialista especificasse o que ele considerava inconsistência através de regras acionadas por *triggers* (gatilhos). Neste trabalho deseja-se avançar mais um passo e diminuir ou até eliminar a necessidade de interferência do especialista no processo, de forma que o próprio computador seja capaz de detectar o que está errado.

Enembreck [23], utiliza a LP em reconhecimento de padrões, no problema da verificação de assinaturas manuscritas. O trabalho de Enembreck [23] trata de diversas técnicas que usam a LP para tratar inconsistências em casos, onde cada caso é representado como uma seção de um reticulado sobre uma assinatura. Dessa forma, a detecção de inconsistências é utilizada para validar a assinatura analisada. Enembreck utiliza algumas técnicas para detectar inconsistências entre os diferentes quadrantes, tais como a aplicação do algoritmo de Naïve Bayes e árvores de decisão justamente para obter os fatores evidenciais.

Outro trabalho importante foi desenvolvido por Dubois *et al*[21] e utiliza lógica fuzzy em RBCs, idéia que se assemelha a usar LP, uma vez que a lógica fuzzy pode ser generalizada pela LP [47]. Apesar de mesclar conceitos semelhantes, o trabalho de Dubois utiliza a lógica fuzzy com objetivo completamente distinto. No trabalho de Dubois, a lógica fuzzy é empregada no processo de recuperação de casos passados, em particular no passo de comparação de um caso analisado contra os casos em memória.

Torres *et al*[76] descreve um sistema híbrido que utiliza LP e Lógica Fuzzy para tomar decisões. Ambas as lógicas são usadas com o objetivo de responder de forma mais precisa e relevante, além de garantir que o sistema não entre em colapso devido a uma possível inconsistência[9]. O sistema tem uma semelhança com este trabalho, já que o RBC é uma ferramenta para tomada de decisões, e estará utilizando a LP para que o RBC incorpore

[9]Uma inconsistência se não tratada, permitirá trivializar os resultados de classificação levando ao colapso lógico da base.

as mesmas características propostas em termos de resposta.

Capítulo 3

Raciocínio Baseado em Casos com Uso de Fatores Evidenciais

O sistema desenvolvido pode ser resumido como um classificador K-Vizinhos que utiliza os fatores evidenciais da PrLE como parte da métrica usada para determinar os vizinhos mais próximos. Naturalmente, por trás desse resumo simples, há muitos aspectos complexos que merecem atenção, e esta seção tem como objetivo apresentar as técnicas e teorias envolvidas nesse trabalho.

3.1 Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma subárea de IA que pesquisa métodos computacionais relacionados à aquisição de novos conhecimentos, novas habilidades e novas formas de organizar o conhecimento já existente [64].

Mitchell [54], define o AM como "Qualquer programa de computador que aumenta sua performance de uma tarefa através da experiência".

Técnicas de AM têm sido muito usadas em todos os ramos da computação, por exemplo, reconhecimento de imagens, sistemas baseados em conhecimento, roteamento de redes e processamento de textos, conseguindo resultados satisfatórios e, às vezes, até melhores do que se esperava. As técnicas de AM são classicamente divididas em técnicas de aprendizado supervisionado e não supervisionado. No aprendizado supervisionado, o conjunto de dados do qual se pretende extrair conhecimento já vem todo rotulado, isto é, a cada

instância está associada sua classificação, a que o algoritmo de AM deve aprender a realizar. No aprendizado não supervisionado, o conjunto de dados não vem rotulado, sendo o algoritmo de AM incumbido de tentar agrupar os dados de acordo com suas características da melhor maneira possível, ou seja, o que se chama de agrupamento *clustering*. As técnicas de AM podem ainda ser classificadas de acordo com o paradigma que seguem, que pode ser simbólico, estatístico, neural ou genético. O aprendizado simbólico se caracteriza por extrair conhecimento que seja acessível e interpretável por seres humanos; o aprendizado estatístico trabalha com fórmulas estatísticas e probabilidades; o aprendizado neural consiste, principalmente, no uso de redes neurais para classificação; o aprendizado genético, por fim, engloba os algoritmos genéticos e suas aplicações.

3.1.1 Aprendizado Bayesiano

O aprendizado bayesiano é do tipo supervisionado, já que são fornecidos ao algoritmo de AM as instâncias juntamente com seus rótulos, ou seja, as classes. Seguindo o paradigma estatístico, o algoritmo faz uso de fórmulas estatísticas e cálculo de probabilidades para realizar a classificação [54]. As vantagens do AM estatístico, especialmente o aprendizado bayesiano, são, principalmente:

- O fato de se poder embutir nas probabilidades calculadas o conhecimento de domínio que se tem;
- A capacidade das classificações feitas pelo algoritmo de AM se basearem em evidências fornecidas, que podem aumentar ou diminuir as probabilidades das classes a serem observadas em uma nova instância que se quer classificar.

3.2 Gerenciamento da Incerteza

3.2.1 Teoria da Probabilidade

É uma aproximação matemática para processar informações incertas. Foi criada por um grupo de jogadores franceses, com o intuito de tornar o jogo menos aleatório. Mais tarde, por volta do ano de 1654, Blaise Pascal e Pierre de Fermat desenvolveram a Teoria da Probabilidade Clássica, usada ainda hoje para extrair inferências numéricas de dados.

Atualmente, pesquizadores de IA utilizam-se da probabilidades para solução de diversos problemas, como manipulação de informações incertas em SE - Sistemas Especialistas [13, 46, 59] e para classificação em sistemas de AM e Data Mining [14, 61].

Uma possível abordagem da TP - Teoria da Probabilidade propõe a existência de um valor *P(E)*: Probabilidade que consiste na possibilidade de ocorrência de um evento E a partir de uma experiência de eventos aleatórios, ou seja, ao realizar-se uma determinada experiência um número considerável de vezes, então a frequência relativa do evento E tende para *P(E)*.

O conjunto de todos os possíveis resultados de uma experiência é denominado *espaço amostral S*.

3.2.2 Teorema Naïve Bayes

Uma abordagem de AM baseada no paradigma probabilístico é o classificador Naïve Bayes [8, 15, 65]. Este classificador pressupõe que a probabilidade de uma evidência conjuntiva $e = (av_i,...av_n)$ pertencer a uma hipótese h é dada pelo produto da probabilidade da ocorrência de cada um dos valores de seus atributos, uma vez que os atributos são considerados independentemente. Apesar da suposição de independência não ser verdadeira para a maioria dos domínios de aplicação do mundo real, onde geralmente há fortes correlações entre os atributos, verifica-se em diversos trabalhos [50, 31] que a classificação produzida pela aplicação do classificador Naïve Bayes apresenta altas taxas de acerto.

Dado um conjunto de treinamento E, formado por exemplos na forma: e=(a_{vi},...,a_{vn}) tal que, a_{vi} é o valor para o atributo a_i, a probabilidade estimada desse conjunto representar uma hipótese h é dada por:

$$p(\mathrm{e}|\mathrm{h}) = p(\mathrm{h} * \prod_{i=1}^{n} P(a_{vi}|\mathrm{h}) \tag{3.1}$$

Utilizando-se a regra de Bayes tem-se:

$$p(\mathrm{h}|\mathrm{e}) = \frac{p(\mathrm{h} * \prod_{i=1}^{n} P(a_{vi}|\mathrm{h})}{\sum_{j=1}^{k} p(h_j * \prod_{i=1}^{n} P(a_{vi}|h_j)} \tag{3.2}$$

A suposição de independência dos atributos feita por Naïve Bayes define o cálculo da probabilidade de uma evidência conjunta $e = (a_v i, ...a_v n)$ como sendo o produto das probabilidades individuais de cada atributo [54], portanto, o modelo aprendido pelo classificador Naïve Bayes é formado pelo conjunto de probabilidades - $p(h)$ e $p(h|e)$ - calculadas a partir dos dados. Pode-se afirmar que a busca por uma determinada hipótese não é realizada através do espaço de hipóteses como em outros métodos de aprendizado, mas calculando-se a frequência de valores do conjunto de treinamento.

Na Seção 4.1 pode-se verificar exemplos da aplicação do classificador Naïve Bayes realizados neste trabalho.

3.3 Arquitetura do Sistema

A apresentação da Arquitetura do Sistema pode ser dividida em duas fases, de treinamento e de classificação. Nesta seção são apresentadas estas arquiteturas, primeiramente do ponto de vista genérico e, em seguida, descrevendo com detalhes os módulos apresentados.

A fase de treinamento engloba as etapas de pré-processamento da base, ou seja, leitura dos dados, discretização e determinação dos fatores evidenciais a serem usados posteriormente no processo de classificação. A arquitetura do sistema na fase de treinamento pode ser vista à Figura 3.3.1, onde:

- MC: representa o fator de crença;
- MD: representa o fator de descrença.

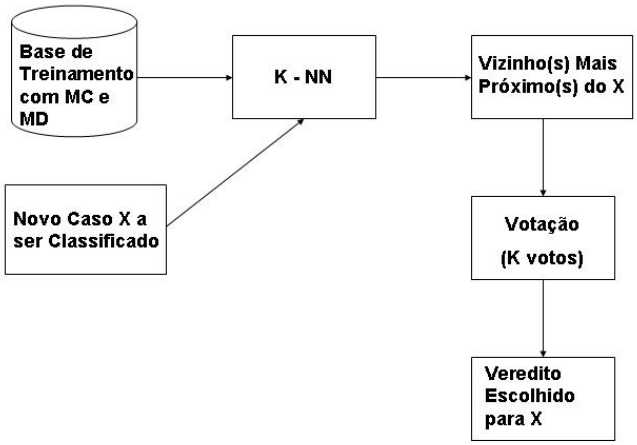
Figura 3.3.1: Ilustração Genérica da Arquitetura do Treinamento.

A fase de classificação supõe que todos os dados necessários para avaliar novos casos já foram obtidos durante o treinamento. Dessa forma, o único e exclusivo objetivo é, dado um novo caso, atribuir a classe mais provável do mesmo. A arquitetura do classificador K-Vizinhos pode ser vista na Figura 3.3.2, onde:

- MC: representa o fator de crença;

- MD: representa o fator de descrença;
- K-NN: representa o classificador K-Vizinhos mais Próximos (*K-Nearest Neighbours*);
- X: representa o novo caso a ser classificado.

Figura 3.3.2: Ilustração Genérica da Arquitetura do Classificador K-Vizinhos.



O sistema descrito a seguir foi totalmente implementado em *Visual C*$^{++}$ da Microsoft.

3.3.1 Fluxo Geral

Pré-Processamento

Inicialmente, dada uma base de casos, o sistema discretiza os valores dos atributos desta base. Em seguida, o sistema utiliza o classificador Naïve Bayes para calcular as probabilidades dos exemplos para cada classe, e a partir destas probabilidades calcula os fatores de crença e descrença para cada caso analisado.

`Algoritmo: pré-processamento`

1. `Lê dados`
2. `Marca os atributos cujo valor é desconhecido`
3. `Discretiza dados`

4. `Calcula os fatores de crença e descrença da base de treinamento`

Classificador

O classificador funciona da seguinte forma: dado um caso novo, é utilizado o algoritmo K-Vizinhos para determinar os vizinhos mais próximos ao caso dado. Em seguida é realizada a votação, que utiliza os fatores de crença e descrença para determinar a provável classe na qual esse caso se encaixa.

`Algoritmo: classificação`

1. `Recebe o caso`

2. `Verifica quais são os vizinhos mais próximos`

3. `Faz a votação (0 a 1)`

A **Votação** consiste em utilizar a classe dos vizinhos mais próximos e combinando a distância e o fator evidencial determinar os votos destes vizinhos para formar o veredito. O veredito é dado pela soma dos votos sendo a classe do caso X determinada pela maior soma dos votos. Este processo será melhor definido na Seção 3.7.

3.4 Características de Entrada de Dados e Representação

3.4.1 Descrição das Bases da Universidade da Califórnia

Foram utilizadas algumas bases de casos disponibilizadas pelo UCI [11] (Repository of Machine Learning Databases), da Universidade da Califórnia. Este é um repositório das bases de dados, das teorias do domínio e dos geradores dos dados que são usados pela comunidade de AM para a análise empírica de algoritmos de AM.

3.4.2 Formato da Base

O formato é disposto da seguinte forma:

Cada caso em uma linha e cada campo está separado por uma vírgula. Foi feito uma extensão neste trabalho para especificar os dois tipos de atributos que podem ser:

- Numérico: é um atributo quantitativo de variável geralmente contínua que necessita ser discretizada.
- Nominal: é um atributo de valor discreto qualitativo ao invés de quantitativo. Por isso, não necessita ser discretizado.

Os campos também podem ser de dois tipos:

- Nominal: é um nome representado por uma *string*[1]. Exemplos:
  - condições climáticas (ensolarado, nublado, chuvoso e ventando);
  - nota por conceito;
  - cor da pele.
- Numérico Contínuo: é um número real. Exemplos:
  - temperatura;
  - velocidade;
  - umidade relativa do ar.

[1]String é uma sequência de caracteres.

3.4.3 Campos Sem Valor Definido

O sistema também leva em conta os campos que não possuem valor definido. O sistema é capaz de lidar com o tratamento de valores desconhecidos do seguinte modo: criou-se um tipo *TFloat* que representa, além do valor do campo, se o valor está definido ou não.

3.5 Discretização

Dado que uma das premissas deste trabalho é que a Base de Casos utilizada não está apropriadamente estruturada, antes de efetuar qualquer análise, é necessário que os dados sejam tratados e readequados. A fase de pré-processamento consiste exatamente desta etapa, na qual os dados são tratados para simplificar o processamento principal do sistema.

Neste trabalho, existem quatro etapas de pré-processamento: 1) como primeira etapa a leitura dos dados da base de casos para uma estrutura de lista; 2) como segunda etapa a marcação dos atributos cujo valor é desconhecido, através de uma variável tipo *TFloat* que representa o valor desconhecido anotado como "?"; 3) como terceira etapa cria-se uma **Tabela de Símbolos** que relaciona os valores dos campos nominais a números. Usando essa tabela, todos os campos nominais tem seus valores substituídos por esses números. Desta forma a implementação do algoritmo tornou-se mais simples e eficientes, pois é mais fácil computacionalmente trabalhar com números do que com *string*. E quando necessário o nome, basta utilizar o número correspondente e procurá-lo na tabela de símbolos; 4) como quarta etapa a discretização dos atributos numéricos contínuos ou atributos numéricos discretos cujo número de valores discretos seja muito grande[2], o que tornaria pouco confiáveis os resultados de distância para o uso do classificador K-Vizinhos.

A Discretização consiste em categorizar atributos numéricos em um conjunto limitado de classes, aumentando a robustez do sistema a ruídos e deficiências nos dados e agrupando valores que, embora diferentes, representam uma mesma condição. Isto é particularmente necessário para a aplicação do Algoritmo Naïve Bayes.

O algoritmo de discretização toma como base aquele apresentado por Mitchell [54], página 72, fazendo algumas modificações para se adequar ao método desenvolvido. O

[2]O número de valores discretos será considerado muito grande sempre que a distância entre valores consecutivos for menor que a largura mínima definida no algoritmo supervisionado de Mitchell modificado, com Largura Mínima de Valor Discreto.

algoritmo é supervisionado, isto é, utiliza uma base de treinamento previamente entrada. Para acompanhar o algoritmo, vale fazer algumas definições:

- O atributo em *discretização* é o atributo que será discretizado.
- A classe é o atributo que representa a classificação de um determinado caso.
- Valor discreto é um valor simbólico que substituirá o valor real dos atributos em discretização.

O objetivo do algoritmo de discretização é determinar valores discretos distintos somente quando necessário.

3.5.1 Algoritmo de Mitchell Modificado

O algoritmo supervisionado de Mitchell modificado, com Largura Mínima de Valor Discreto é como a seguir:

1. ordene todos os casos, do menor para o maior, segundo o atributo a ser discretizado;
2. Calcule a largura mínima de cada faixa, da seguinte forma:

$$LargMinima \leftarrow \frac{(MAX - MIN)}{nLinhas} * M \tag{3.3}$$

   Onde:

   - *LargMinima*: é a menor largura que se aceita para as faixas discretizadas;
   - MAX: maior valor que atributo assume;
   - MIN: menor valor que atributo assume;
   - *nLinhas*: número total de casos;
   - M: é um multiplicador para evitar que a faixa mínima fique estreita demais. Neste trabalho o valor de M é 5;

3. escolha um nome N distinto a ser utilizado como valor simbólico do atributo;

4. faça P_i (limite inferior do intervalo) ser igual ao valor numérico do atributo escolhido no primeiro caso dividido por dois;

5. enquanto houverem casos faça:

   (a) Seja $V_{anterior}$ o valor numérico do atributo do caso anterior

   (b) Seja V_{atual} o valor numérico do atributo do caso atual

   (c) Se o valor das classes do caso atual e do caso anterior são diferentes:

   - Calcule $PQ = \frac{V_{anterior}+V_{atual}}{2}$
     (PQ: candidato a ser o ponto final da faixa e também é o ponto médio.)
   - Calcule $Delta$ = PQ - P_i
     ($Delta$ é o tamanho da faixa do valor discreto atual.)
   - Se $Delta >= LargMínima$ então:
     - Calcule $PM = PQ$
     - Faça com que N represente o intervalo $[P_i, PM[$
     - Faça a atribuição $P_i = PM$
     - troque o valor de N para que seja algum N arbitrário diferente dos valores utilizados

   (d) faça com que o valor do atributo atual seja N

6. Faça com que N (a última classe de valor criada) represente o intervalo $[P_i, V_{atual}]$

No algoritmo concebido em Mitchell [54], a idéia básica é que se uma sequência de casos é encontrada para a qual o atributo de classificação permanece inalterado, isso indica que todos esses casos deveriam receber o mesmo valor discreto para este atributo. Dessa forma, a idéia é escolher um valor discreto arbitrário e trocá-lo sempre que um par de casos consecutivos é encontrado para os quais a classificação é diferente.

Observe que esse critério assume que o valor do atributo está diretamente relacionado ao valor da classe e portanto classes diferentes implicam que o valor do atributo deve ser diferente também. Isso é um problema quando se tenta detectar inconsistências, pois sabe-se que se existe inconsistência, esta não tem necessariamente de seguir qualquer padrão (por exemplo, o padrão de que o atributo está relacionado com a classe) e portanto a

própria discretização estaria sendo afetada pela inconsistência. De fato, o maior problema é que a discretização esconde algumas inconsistências.

Levando-se em consideração a inconsistência inter-casos, foi observado anteriormente na Seção 2.3 que essa inconsistência é detectada quando casos com atributos iguais ou que seguem uma determinada regra resultam em conclusões (entenda conclusão como a classificação) diferentes em situações em que deveriam ter a mesma conclusão. Retornando agora ao atributo que está sendo discretizado e supondo que este é um dos atributos importantes para uma determinada conclusão quando numericamente semelhantes. Durante o processo de discretização, eventualmente serão encontrados casos inconsistentes porque existem atributos semelhantes, cuja classe é diferente. Supondo-se que esses casos são consecutivos na ordenação, pelo critério de Mitchell, como as classes são diferentes, o algoritmo de discretização vai atribuir valores discretos diferentes (mesmo que o atributo em discretização seja numericamente igual nos dois casos). Ou seja, como os valores são diferentes depois da discretização, e não será mais possível correlacionar os dois casos e perceber a inconsistência (mesmo para um ser humano). É inadmissível perder essa informação, quando o principal foco deste trabalho é detectar e tratar as inconsistências.

Dessa forma, neste trabalho optou-se por relaxar a regra de que a mudança da classe necessariamente implica em valores discretos diferentes. Para atingir este fim, adiciona-se ainda que além dos valores de classes serem diferentes, é também necessário que o valor real do atributo em discretização seja consideravelmente diferente. Para expressar essa diferença em termos objetivos, o resultado do módulo da subtração dos valores deve ser maior do que uma largura mínima calculada da seguinte forma:

$$\frac{(\text{Maior Valor Possível} - \textit{Menor Valor Possível})}{\textit{Número de Casos}} \times M \tag{3.4}$$

A constante *(M)* foi um valor arbitrariamente escolhido (5 nesse trabalho), para evitar que a largura ficasse demasiadamente estreita. Não foi realizado nenhum experimento para verificar qual a melhor constante a ser utilizada.

3.5.2 Outros Métodos de Discretização Testados

Outros três métodos de discretização não supervisionados também foram testados durante a realização deste trabalho.

1. Número Fixo de Classes:

   A discretização por Número Fixo de Classes determina uma quantia fixa de valores discretos (classes) a serem criados, independente da quantidade de casos da base. Supondo que sejam criadas N_c classes, cada classe deverá cobrir aproximadamente o mesmo número de casos, ou seja, é a quantia total de casos dividido pelo número de classes a serem criadas. A idéia é bastante simples, de modo que cada atributo tem seu valor alterado de acordo com o seguinte procedimento:

   - Dado um caso C_i e o atributo $A(C_i)$ a ser discretizado.
   - Seja $V_{\max}$ o maior valor que o atributo A_i pode assumir.
   - Seja $V_{\min}$ o menor valor que o atributo A_i pode assumir.
   - Seja N_{casos} a quantia de casos da base.
   - $Delta \leftarrow \frac{V_{\max} - V_{\min}}{N_{\text{casos}}}$
   - Para todo caso C_i, $A(C_i) \leftarrow \frac{A(C_i) - V_{\min}}{Delta}$

   Resumindo, o processo consiste em ordenar os casos e fazer com que os casos sejam distribuídos em N_c partições contíguas. Observa-se que o número de classes a serem criadas é fixo e independente da base sendo processada.

2. Largura Constante:

   Outra forma bastante simples de discretização é a realizada com Largura Constante. Nessa modalidade de discretização, são criadas faixas discretas de tamanho fixo. Dessa forma, todos os casos que têm o atributo a ser discretizado dentro da faixa em questão, terá o atributo com valor discreto correspondente ao da classe vinculada à faixa de valores. A largura é fixada inicialmente e não muda de uma base de casos para outra.

3. Frequência Constante:

   A discretização por Frequência Constante consiste de um processo bastante simples, no qual parte-se da premissa de que cada valor discreto deve ter o mesmo número de casos, mas sem a necessidade de fixar um número máximo de valores discretos. Os casos são ordenados também. A idéia é percorrer os casos atribuindo um mesmo valor discreto para o atributo em questão. A partir do momento que o valor discreto (classe) consiste de um conjunto de tamanho L, onde L é a largura fixada, é arbitrado um novo valor para atribuir aos próximos casos.

   Mesmo sendo um método inadequado, este método ainda é melhor do que os anteriores porque ao menos considera a faixa de valores do atributo sendo discretizado.

Desvantagens destes Métodos

Os três métodos de discretização: Número Fixo de Classes, Largura Constante e Frequência Constante sofrem das seguintes desvantagens:

1. O critério de atribuição de valor discreto desconsidera as características de cada base de casos e do próprio atributo em discretização. Dessa forma, para cada base seria necessário a intervenção de um ser humano para ajustar os parâmetros (tais como largura, frequência, número de valores discretos) de modo a obter bons resultados.

2. Os métodos desconsideram a possibilidade do valor do atributo ter alguma relação com a classificação do caso. Essa distorção praticamente elimina qualquer semântica, mesmo implícita, que o atributo possua.

3. O fato dos atributos serem discretizados sem considerar uma possível relação com a classificação torna sem sentido a busca de inconsistência baseado nesse atributo. Naturalmente, a base discretizada teria pouca utilidade para a proposta desse trabalho.

Desta forma optou-se pelo uso unicamente do método de Mitchell Modificado, que também foi o que produziu os melhores resultados nos experimentos preliminares realizados.

3.6 Fatores Evidenciais

O método de K-Vizinhos mais próximos tradicional considera que cada caso votante acredita em sua própria classe, com grau de crença de 100%. É interessante calcular com que probabilidade a classe do vizinho votante está correta, e deste modo levar em conta os fatores evidenciais de crença e descrença do vizinho votante em sua própria classe, quando da consideração de seu voto.

A elaboração do algoritmo de determinação de fatores evidenciais considera inicialmente a utilização do algoritmo Naïve Bayes. O cálculo dos fatores evidenciais, descrito em Enembreck [23], consiste na verdade de vários estágios para que se utilizem todos os dados.

O cálculo de cada probabilidade dentro de uma classe é simples: supondo que o caso foi classificado dentro de uma determinada classe, divida o número de ocorrências de um determinado valor para o atributo considerado pelo número de casos com mesma classificação.

$$\begin{aligned} P(\text{Caso}|\text{Classe}) = & \\ & P(Atributo1|\text{Classe})\times \\ & P(Atributo2|\text{Classe})\times \\ & P(Atributo3|\text{Classe})\times \\ & ... \\ & P(Atributon|\text{Classe}) \end{aligned} \tag{3.5}$$

A equação anterior deverá ser aplicada para cada caso selecionado.

Uma vez que foram determinadas as probabilidades de uma classe ocorrer dados os atributos do caso considerado, é necessário calcular a probabilidade de cada caso estar correto se for fixada a classe.

$$P(\text{Classe}|\text{Caso}) = \frac{P(\text{Classe}) \times P(\text{Caso}|\text{Classe})}{\begin{array}{c} P(\text{Caso}|\text{Classe1}) \times P(\text{Classe1})+ \\ P(\text{Caso}|\text{Classe2}) \times P(\text{Classe2})+ \\ P(\text{Caso}|\text{Classe3}) \times P(\text{Classe3})+... \\ P(\text{Caso}|\text{Classen}) \times P(\text{Classen}) \end{array}} \tag{3.6}$$

Finalmente, a partir das probabilidades calculadas, pode-se calcular os valores de crença e descrença de cada caso (*MC* e *MD*), respectivamente, usando as equações que foram introduzidas pelo sistema MYCIN [13, 56], e também utilizadas em [23], como apresenta-se abaixo:

$$MC[Classe, Caso] = \begin{cases} 1 & \text{se p(Classe)} = 1 \\ \frac{max[p(Classe|Caso), p(Classe)] - p(Classe)}{1 - p(Classe)} & \textit{caso contrário} \end{cases} \tag{3.7}$$

$$MD[Classe, Caso] = \begin{cases} 1 & \text{se p(Classe)} = 0 \\ \frac{min[p(Classe|Caso), p(Classe)] - p(Classe)}{-p(Classe)} & \textit{caso contrário} \end{cases} \tag{3.8}$$

Onde:

- *MC[Classe,Caso]*: é o aumento proporcional da crença no resultado de *Classe* do *Caso* em relação à crença da classe na base como um todo;
- *MD[Classe,Caso]*: é a diminuição proporcional da crença no resultado de *Classe* do *Caso* em relação a descrença da *classe* na base como um todo;
- *p(Classe)*: é a probabilidade de ocorrência da *classe* na base como um todo, a qual representa a crença da *Classe*;
- 1-*p(Classe)*: é a probabilidade de ocorrência de *Classes* diferentes da classe considerada em toda base.

3.7 O Classificador K-Vizinhos

O classificador K-Vizinhos usa apenas as instâncias existentes do problema. Existem trabalhos que afirmam que a complexidade de um algoritmo do estilo K-NN é o $O(m\ n)$, onde m é o número de atributos e n é o número de casos da base [77]. Essa complexidade pode ser ainda diminuida com algumas técnicas como "Vizinhos Aproximativos", introduzindo mais informação na base de casos [34]. Sua aplicação fundamenta-se em duas partes: métrica da distância e forma de cálculo dos votos.

A base de treinamento é meramente formada por instâncias de casos, e o classificador associa a um novo caso a mesma classe que a das K-instâncias mais próximas.

Um exemplo da aplicação do classificador K-Vizinhos:

Dado um caso C onde tem-se 5 vizinhos $(v_1, v_2, ..., v_5)$. Os vizinhos são casos da base de treinamento. Cada caso da base de treinamento tem uma determinada classificação. O objetivo é achar a classificação de C.

O algoritmo usual de K-Vizinhos simplesmente verifica qual a classificação mais frequente nos vizinhos e atribui a C. É como se cada vizinho votasse em sua própria classificação para definir a classe de C.

A divisão do espaço de representação para o K não aparece na fórmula. Uma indicação para se obter o número K utiliza a fórmula:

$$K \cong \sqrt{\frac{M}{C}} \tag{3.9}$$

Onde: $\frac{M}{C}$ representa o número médio de pontos de aprendizagem por classe [16].

No algoritmo atual ao invés de fazer um voto comum, a crença do caso associado ao vizinho também é considerada. Ao invés de ter um voto com valor 1, o vizinho tem um voto com valor proporcional ao valor de sua crença, em função dos valores de MC e MD.

No final o veredito é dado pela soma dos votos para cada classe, sendo a classe do caso avaliado aquela com maior soma de votos.

3.7.1 Métrica da Distância

O algoritmo utiliza uma métrica fixa para calcular a distância, e essa distância determina quais os vizinhos a serem considerados pelo processo de classificação. Uma vez que os K-

Vizinhos são obtidos, ainda resta considerar como suas respectivas classificações deverão ser utilizadas para determinar a classe do caso em questão.

A métrica da distância é calculada da seguinte forma:

Dados dois casos C_a e C_b de uma mesma base. Ainda, seja $C[i]$ o i-ésimo atributo do caso C. A distância total d entre C_a e C_b pode ser computada da seguinte forma:

$$d(C_a, C_b) = \sum_{i=0}^{n} da_i \tag{3.10}$$

onde da_i pode ser visto como a distância entre os i-ésimos atributos de C_a e C_b, de modo que:

$$da_i = \begin{cases} 0 & \text{se } C_a[i] = C_b[i] \\ 1 & \text{caso contrário} \end{cases} \tag{3.11}$$

Dessa forma, quanto mais "parecidos" forem os casos, menor a distância.

A métrica da distância fixa entre atributos diferentes foi escolhida porque, uma vez discretizados todos os atributos contínuos, tais valores discretizados passam a ser considerados simbólicos, e naturalmente não haverá uma escala de valores entre eles, uma vez que o algoritmo tem aplicação genérica, e não há nenhum especialista para julgar uma escala de valores entre as faixas discretizadas.

3.7.2 Descrição do Classificador e Fórmulas de Votação

Durante a classificação, a classe de um determinado vizinho deve ser considerada no processo. Cada vizinho tem direito a um "voto"em sua própria classe como sendo a classe do caso a ser classificado. O peso do voto pode ser influenciado pela distância e pelos fatores evidenciais. Os passos do algoritmo classificador de K-Vizinhos são:

1. escolhe os K-Vizinhos mais próximos;

2. faz a Votação, que consiste em utilizar a classe dos vizinhos mais próximos e, combinando a distância e o fator evidencial, determina qual é a classe do caso avaliado;

3. obtém o Veredito de classe, somando os pesos dos votos de cada classe, e tomando como veredito (classe escolhida) a classe com maior soma.

Foram escolhidas seis fórmulas para calcular o peso do voto de cada vizinho, onde:

- `peso_do_voto`: representa o valor do peso do voto;
- MC: representa o fator de crença;
- MD: representa o fator de descrença;
- *distância*: é um parâmetro que representa o quanto diferem os atributos entre dois casos, de modo que casos mais próximos (menor distância), apresentam maior coincidência de valores de atriburos discretizados e casos mais distantes (maior distância), tem menos valores de atributos discretizados em comum.

1. Fórmula 1:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = MC \tag{3.12}$$

2. Fórmula 2:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = \frac{1}{\textit{distância}} \tag{3.13}$$

3. Fórmula 3:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = MC * (1 - MD) \tag{3.14}$$

4. Fórmula 4:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = \frac{1}{((1 - MC) * MD * \textit{distância})} \tag{3.15}$$

5. Fórmula 5:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = MC * (1 - MD) * \frac{1}{\textit{distância}} \tag{3.16}$$

6. Fórmula 6:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = 1 \tag{3.17}$$

3.7.3 Considerações sobre as Fórmulas de Votação

Algumas considerações são feitas em relação as fórmulas de votação escolhidas:

1. Fórmula 1:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = MC \tag{3.18}$$

   - considera como fator evidencial da classe somente o grau de crença da classe proposta pelo vizinho votante;
   - não leva em conta o grau de descrença;
   - a decisão é dada por um fator da PrLE, o que confere grande peso a esta ferramenta, pois as distâncias são levadas em conta apenas para determinar quem são os vizinhos mais próximos, ou seja, os votantes da classificação;
   - de maneira similar ao processo cognitivo humano, o votante com grau de crença=0 em sua própria classe vota em branco.

2. Fórmula 2:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = \frac{1}{\textit{distância}} \tag{3.19}$$

   - considera somente o quão próximo o vizinho votante está do caso em teste;
   - com o uso desta fórmula não está sendo levada em consideração a PrLE;
   - é importante observar que o peso do voto e inversamente proporcional à distância;
   - no caso de coincidência total de atributos discretizados (distância=0), o algoritmo irá atribuir o valor arbitrário de 0.0001 para a distância. Deste modo o valor do peso do voto será 10000.00, o que confere peso decisivo para as coincidências, melhorando a detecção de inconsistências: caso as classes de dois casos coincidentes não sejam as mesmas, então a inconsistência é óbvia.

3. Fórmula 3:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = MC * (1 - MD) \tag{3.20}$$

   - considera tanto o fator de crença como o de descrença para composição do peso do voto;

- o peso do voto é diretamente proporcional ao grau de crença e ao complemento do grau de descrença;
- de maneira similar a Fórmula 1 utiliza-se da LP para decidir o voto, e das distâncias apenas para escolher os vizinhos votantes;
- da mesma maneira que na Fórmula 1, o vizinho com grau de crença=0 ou grau de descrença=1 votará em branco.

4. Fórmula 4:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = \frac{1}{((1 - MC) * MD * \textit{distância})} \tag{3.21}$$

- esta fórmula leva em conta tanto os fatores evidenciais da LP quanto a distância entre o caso em teste e o caso votante;
- considerou-se o peso do voto como inversamente proporcional ao grau de descrença, ao complemento do grau de crença e a distância;
- nesta fórmula, caso o grau de descrença seja=0, ou o grau de crença seja=1, ou a distância=0, o denominador será substituído por 0.0001. Desta maneira tanto uma coincidência total de valores discretizados dos atributos, quanto um grau de crença de 100% quanto um grau de descrença nulo, terá um peso decisivo no veredito, tornando o voto de qualquer vizinho com fatores evidenciais diferentes destes pouco influente no resultado.

5. Fórmula 5:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = MC * (1 - MD) * \frac{1}{\textit{distância}} \tag{3.22}$$

- nesta fórmula, o peso do voto será diretamente proporcional ao grau de crença e ao complemento do grau de descrença, e inversamente proporcional a distância;
- esta fórmula procura combinar todos os fatores evidenciais da LP e da distância, porém, neste caso somente a coincidência total de valores de atributos discretizados (distância=0) terá um peso decisivo na votação.

6. Fórmula 6:

$$\texttt{peso\_do\_voto}(\textit{Caso, Classe do Caso}) = 1 \tag{3.23}$$

   - nesta fórmula, o peso do voto será 1;
   - não utiliza a LP;
   - equivale ao uso do classificador K-Vizinhos tradicional;
   - fórmula testada somente para comparação de resultados: uso da LP versus não uso da LP.

Capítulo 4

Cenário de Experimentação

4.1 Ambiente de Testes

Esta seção tem como objetivo ilustrar a aplicação das técnicas desenvolvidas nesse trabalho. Foi criado um exemplo composto de uma base de casos artificial, e para esta são apresentados os passos executados no cálculo dos fatores evidenciais e a aplicação do classificador K-Vizinhos.

Considera-se o seguinte cenário: uma empresa deseja fazer uma auditoria no processo de recrutamento utilizado pelo setor de recursos humanos. O objetivo é determinar se os critérios de avaliação foram corretamente aplicados, o que implica determinar se a avaliação foi justa. Para fazer essa auditoria, a empresa dispõe de uma base de casos, denominada de Candidato, na qual cada coluna representa um critério e o valor atribuído ao mesmo, além da coluna que corresponde ao veredito, aprovado, reprovado ou reavaliação.

A base de casos fictícia Candidato possui as seguintes características:

1. Número de Instâncias: 25
2. Número de Atributos: 5 mais o atributo de classe que é nominal
    - 2 valores contínuos
    - 3 de valores nominais (ou simbólicos)
3. Informação sobre os atributos: (Nome do Atributo/Valores Possíveis)

(1) Cod: Código, somente utilizado para identificação e não será considerado no processamento

(2) CA: Currículo Acadêmico, avaliado por conceito que vai de A até F

(3) CP: Currículo Profissional, avaliado por conceito que vai de A até F

(4) Experiência: Experiência na Área que o candidato possui. Avaliada pelo número de anos, que vai de 0 até 10+

(5) Teste: é uma prova com resultado de notas entre 0 e 10.0

(6) Entrevista: avaliada por conceito que pode ser Muito Bom, Bom, Fraco ou Regular

(7) Veredito: pode ser Aprovado, Reprovado ou Reavaliacao

4. Valores de Atributos Faltantes: nenhum

A base Candidato original pode ser vista na Tabela 4.1.1.

Divididiu-se a base de casos Candidato da seguinte forma: 70% do total de casos para a base de Treinamento e 30% para a base de Testes.

O critério de divisão segue a ordem de entrada dos casos na base, ou seja, os 70% primeiros casos farão parte da base de Treinamento e os 30% últimos farão parte da base de Testes. Neste exemplo explicativo não foi realizada nenhuma tentativa de manter a distribuição de frequência de classes da base original nas bases de Treinamento e Testes.

O algoritmo de discretização foi aplicado apenas na base de Treinamento obtendo-se as faixas de valores discretos. Em seguida, cada valor contínuo dos atributos da base de Testes foi discretizado de acordo com as faixas de valor discreto da base de Treinamento.

Após a aplicação do algoritmo de discretização sobre os atributos contínuos da base de treinamento, os seguintes valores foram criados:

- Valores para Experiência na Área:

Coluna	Valor Discreto	Faixa Representada
2	0	[0.50, 6.50[
2	1	[6.50, 9.50[
2	2	[9.50, 10.00[

Tabela 4.1.1: Base Candidato Original

Cod	CA	CP	Experiencia	Teste	Entrevista	Veredito
0	F	F	2.00	1.20	Regular	Reprovado
1	E	E	5.00	4.50	Fraco	Reprovado
2	A	A	9.00	10.00	Muito Bom	Reprovado
3	E	E	5.00	6.40	Fraco	Reprovado
4	F	F	2.00	1.50	Regular	Aprovado
5	A	A	6.00	10.00	Regular	Aprovado
6	C	D	7.00	7.20	Bom	Reavaliacao
7	B	C	5.00	7.80	Muito Bom	Reprovado
8	C	C	8.00	8.10	Bom	Reavaliacao
9	A	A	10.00	9.95	Muito Bom	Aprovado
10	B	B	9.00	8.50	Muito Bom	Aprovado
11	E	E	5.00	4.80	Fraco	Reprovado
12	E	E	6.00	5.50	Fraco	Reprovado
13	A	B	10.00	9.10	Muito Bom	Aprovado
14	B	A	9.00	8.70	Muito Bom	Aprovado
15	E	E	5.00	4.70	Fraco	Reprovado
16	F	F	3.00	8.00	Regular	Reprovado
17	F	F	1.00	4.40	Regular	Reprovado
18	C	C	8.00	8.30	Bom	Reavaliacao
19	D	D	7.00	6.50	Bom	Reavaliacao
20	C	E	4.00	9.50	Bom	Reprovado
21	F	F	2.00	2.40	Regular	Reprovado
22	D	C	8.00	7.40	Bom	Reavaliacao
23	A	A	10.00	9.80	Muito Bom	Aprovado
24	F	C	8.00	3.10	Fraco	Reavaliacao

- Valores para Teste:

Coluna	Valor Discreto	Faixa Representada
3	0	[0.60, 6.45[
3	1	[6.45, 9.10[
3	2	[9.10, 10.00[

A base de casos Candidato discretizada resultante pode ser vista na Tabela 4.1.2, e as probabilidades de cada valor em cada classe pode ser vista na Tabela 4.1.3. Onde: ? significa indefinido. Embora a base Candidato tenha sido simulada artificialmente e por isso não possui nenhum valor indefinido, pelo fato do sistema ser genérico tornou-se necessário considerar possibilidade de valor indefinido.

Tabela 4.1.2: Base de Casos Candidato Discretizada

Base de Treinamento						
Cod	**CA**	**CP**	**Experiencia**	**Teste**	**Entrevista**	**Veredito**
0	1.0000(F)	1.0000(F)	0.0000	0.0000	1.0000(Regular)	1.0000(Reprovado)
1	1.0000(F)	1.0000(F)	0.0000	0.0000	1.0000(Regular)	2.0000(Aprovado)
2	1.0000(F)	1.0000(F)	0.0000	0.0000	1.0000(Regular)	1.0000(Reprovado)
3	1.0000(F)	1.0000(F)	0.0000	0.0000	1.0000(Regular)	1.0000(Reprovado)
4	2.0000(E)	2.0000(E)	0.0000	0.0000	2.0000(Fraco)	1.0000(Reprovado)
5	2.0000(E)	2.0000(E)	0.0000	0.0000	2.0000(Fraco)	1.0000(Reprovado)
6	2.0000(E)	2.0000(E)	0.0000	0.0000	2.0000(Fraco)	1.0000(Reprovado)
7	2.0000(E)	2.0000(E)	0.0000	0.0000	2.0000(Fraco)	1.0000(Reprovado)
8	6.0000(D)	4.0000(D)	1.0000	1.0000	4.0000(Bom)	3.0000(Reavaliacao)
9	4.0000(C)	4.0000(D)	1.0000	1.0000	4.0000(Bom)	3.0000(Reavaliacao)
10	5.0000(B)	5.0000(C)	0.0000	1.0000	3.0000(Muito Bom)	1.0000(Reprovado)
11	1.0000(F)	1.0000(F)	0.0000	1.0000	1.0000(Regular)	1.0000(Reprovado)
12	4.0000(C)	5.0000(C)	1.0000	1.0000	4.0000(Bom)	3.0000(Reavaliacao)
13	5.0000(B)	3.0000(A)	1.0000	1.0000	3.0000(Muito Bom)	2.0000(Aprovado)
14	4.0000(C)	2.0000(E)	0.0000	2.0000	4.0000(Bom)	1.0000(Reprovado)
15	3.0000(A)	3.0000(A)	2.0000	2.0000	3.0000(Muito Bom)	2.0000(Aprovado)
16	3.0000(A)	3.0000(A)	1.0000	2.0000	3.0000(Muito Bom)	1.0000(Reprovado)
Base de Testes						
0	3.0000(A)	3.0000(A)	0.0000	2.0000	1.0000(Regular)	2.0000(Aprovado)
1	4.0000(C)	5.0000(C)	1.0000	1.0000	4.0000(Bom)	3.0000(Reavaliacao)
2	3.0000(A)	6.0000(B)	2.0000	1.0000	3.0000(Muito Bom)	2.0000(Aprovado)
3	3.0000(A)	3.0000(A)	2.0000	2.0000	3.0000(Muito Bom)	2.0000(Aprovado)
4	5.0000(B)	6.0000(B)	1.0000	1.0000	3.0000(Muito Bom)	2.0000(Aprovado)
5	6.0000(D)	5.0000(C)	1.0000	1.0000	4.0000(Bom)	3.0000(Reavaliacao)
6	2.0000(E)	2.0000(E)	0.0000	0.0000	2.0000(Fraco)	1.0000(Reprovado)
7	1.0000(F)	5.0000(C)	1.0000	0.0000	2.0000(Fraco)	3.0000(Reavaliacao)

Após a discretização alguns dos atributos passam a ser representados por valores:

1. Para os atributos CA (Currículo Acadêmico) e CP (Currículo Profissional):
    - 0: representa Indefinido;
    - 1: representa F;
    - 2: representa E;
    - 3: representa A;
    - 4: representa D;
    - 5: representa C;
    - 6: representa B.
2. Para o atributo Entrevista:
    - 0: representa Indefinido;
    - 1: Regular;
    - 2: Fraco;
    - 3: Muito Bom;
    - 4: Bom.
3. Para o atributo Veredito (classe):
    - 0: representa Indefinido;
    - 1: Reprovado;
    - 2: Aprovado;
    - 3: Reavaliação.

Tabela 4.1.3: Probabilidades de Cada Valor em Cada Classe

Probabilidades	**Aprovado**	**Reprovado**	**Reavaliacao**
P(CA=?|Classe)	0.0000	0.0000	0.0000
P(CA=A|Classe)	0.3333	0.0909	0.0000
P(CA=B|Classe)	0.3333	0.0909	0.0000
P(CA=C|Classe)	0.0000	0.0909	0.6667
P(CA=D|Classe)	0.0000	0.0000	0.3333
P(CA=E|Classe)	0.0000	0.3636	0.0000
P(CA=F|Classe)	0.3333	0.3636	0.0000
P(CP=?|Classe)	0.0000	0.0000	0.0000
P(CP=A|Classe)	0.6667	0.0909	0.0000
P(CP=B|Classe)	0.0000	0.0000	0.0000
P(CP=C|Classe)	0.0000	0.0909	0.3333
P(CP=D|Classe)	0.0000	0.0000	0.6667
P(CP=E|Classe)	0.0000	0.4545	0.0000
P(CP=F|Classe)	0.3333	0.3636	0.0000
P(Experiencia=0|Classe)	0.3333	0.9091	0.0000
P(Experiencia=1|Classe)	0.3333	0.0909	1.0000
P(Experiencia=2|Classe)	0.3333	0.0000	0.0000
P(Teste=0|Classe)	0.3333	0.6364	0.0000
P(Teste=1|Classe)	0.3333	0.1818	1.0000
P(Teste=2|Classe)	0.3333	0.1818	0.0000
P(Entrevista=?|Classe)	0.0000	0.0000	0.0000
P(Entrevista=Fraco|Classe)	0.0000	0.3636	0.0000
P(Entrevista=Regular|Classe)	0.3333	0.3636	0.0000
P(Entrevista=Bom|Classe)	0.0000	0.0909	1.0000
P(Entrevista=Muito Bom|Classe)	0.6667	0.1818	0.0000
P(Classe)	0.1765	0.6471	0.1765

4.1.1 Cálculo dos Fatores Evidenciais

Uma vez que a base está discretizada, o sistema deve calcular os fatores evidenciais de cada caso. O cálculo dos fatores evidenciais tem diversos passos. Tomando-se como exemplo o Candidato0 da Base Candidato Discretizada, e substituindo-se os valores das fórmulas anteriormente descritas na Seção 3.6 para adaptar-se a este exemplo, os passos são os seguintes:

1. Calcular as probabilidades de ocorrências de cada classe na base como um todo. No

exemplo do Candidato0 da Base de Treinamento a classe é Reprovado, sendo:

$$\begin{aligned} P(\text{Reprovado}) &= \frac{\text{Número de Ocorrências}}{\text{Número de Casos}} \\ P(\text{Reprovado}) &= \frac{11}{17} \\ P(\text{Reprovado}) &= 0.6471 \end{aligned} \tag{4.1}$$

As probabilidades de cada classe ocorrer na base estão indicadas na última linha da Tabela 4.1.3.

2. Utilizando a base de treinamento calcular a probabilidade de cada valor ocorrer em cada classe:

   No caso do Candidato0, deve-se calcular a probabilidade correspondente a cada valor de atributo que ocorre neste caso, combinada com a Classe do Caso. Por exemplo:

$$\begin{aligned} P(CA = F|\text{Reprovado}) &= \frac{\text{Número de Atributos F em Reprovados}}{\text{Número Total de Reprovados}} \\ P(CA = F|\text{Reprovado}) &= \frac{4}{11} = 0.3636 \end{aligned} \tag{4.2}$$

$$\begin{aligned} P(CP = F|\text{Reprovado}) &= \frac{\text{Número de Atributos F em Reprovados}}{\text{Número Total de Reprovados}} \\ P(CP = F|\text{Reprovado}) &= \frac{4}{11} = 0.3636 \end{aligned} \tag{4.3}$$

$$\begin{aligned} P(Experiencia = 0|\text{Reprovado}) &= \frac{\text{Número de Atributos 0 em Reprovados}}{\text{Número Total de Reprovados}} \\ P(Experiencia = 0|\text{Reprovado}) &= \frac{10}{11} = 0.9091 \end{aligned} \tag{4.4}$$

$$\begin{aligned} P(Teste = 0|\text{Reprovado}) &= \frac{\text{Número de Atributos 0 em Reprovados}}{\text{Número Total de Reprovados}} \\ P(Teste = 0|\text{Reprovado}) &= \frac{7}{11} = 0.6364 \end{aligned} \tag{4.5}$$

Tabela 4.1.4: Probabilidade de Cada Candidato Ocorrer em Cada Classe.

Candidato	**Classe ?**	**Classe Aprovado**	**Classe Reprovado**	**Classe Reavaliação**
0	0.00000000	0.00411523	0.02781728	0.00000000
1	0.00000000	0.00411523	0.02781728	0.00000000
2	0.00000000	0.00411523	0.02781728	0.00000000
3	0.00000000	0.00000000	0.02781728	0.00000000
4	0.00000000	0.00000000	0.03477159	0.00000000
5	0.00000000	0.00000000	0.03477159	0.00000000
6	0.00000000	0.00000000	0.03477159	0.00000000
7	0.00000000	0.00000000	0.03477159	0.00000000
8	0.00000000	0.00000000	0.00000000	0.22222222
9	0.00000000	0.00000000	0.00000000	0.44444444
10	0.00000000	0.00000000	0.00024837	0.00000000
11	0.00000000	0.00411523	0.00794779	0.00000000
12	0.00000000	0.00000000	0.00001242	0.22222222
13	0.00000000	0.01646091	0.00002484	0.22222222
14	0.00000000	0.00000000	0.00062092	0.00000000
15	0.00000000	0.01646091	0.00000000	0.00000000
16	0.00000000	0.01646091	0.00002484	0.00000000

$$P(Entrevista = 1|\text{Reprovado}) = \frac{\text{Número de Atributos 1 em Reprovados}}{\text{Número Total de Reprovados}}$$
$$P(Entrevista = 1|\text{Reprovado}) = \frac{4}{11} = 0.3636 \tag{4.6}$$

A Tabela 4.1.4 indica a probabilidade de cada valor ocorrer em cada classe, e foi construída com essa metodologia.

3. Calcular a probabilidade de cada candidato ocorrer para cada classe. Para o Candidato0 na base de treinamento:

$$\begin{aligned} P(0|\text{Reprovado}) = \\ & P(CA = F|\text{Reprovado})\times \\ & P(CP = F|\text{Reprovado})\times \\ & P(Experiencia = 0|\text{Reprovado})\times \\ & P(Teste = 0|\text{Reprovado})\times \\ & P(Entrevista = Regular|\text{Reprovado}) \end{aligned} \tag{4.7}$$

Substituindo-se pelos valores tem-se:

$$\begin{aligned} P(0|\text{Reprovado}) = 0.3636 * 0.3636 * 0.9091 * 0.6364 * 0.3636 \\ P(0|\text{Reprovado}) = 0.02781728 \end{aligned} \tag{4.8}$$

4. Calcular a probabilidade de cada classe ser verdadeira para cada candidato (uma vez que se procura por inconsistências, não se pode confiar plenamente na classificação original). No caso do Candidato0 utilizando-se a base de treinamento calcula-se:

$$\begin{aligned} P(\text{Reprovado}|0) = \frac{P(\text{Reprovado}) \times P(0|\text{Reprovado})}{} \\ P(0|\text{Aprovado}) \times P(\text{Aprovado})+ \\ P(0|\text{Reprovado}) \times P(\text{Reprovado})+ \\ P(0|\text{Reavaliacao}) \times P(\text{Reavaliacao}) \end{aligned} \tag{4.9}$$

Substituindo-se pelos valores tem-se:

$$\begin{aligned} P(\text{Reprovado}|0) = \frac{0.6471*0.02781728}{(0.00411523*0.1765)+(0.02781728*0.6471)+(0.00000000*0.1765)} \\ P(\text{Reprovado}|0) = 0.9612 \end{aligned} \tag{4.10}$$

Tabela 4.1.5: Probabilidade de Cada Classe ser Verdadeira para Cada Candidato.

Candidato	**Classe 0**	**Classe 1**	**Classe 2**	**Classe 3**
0	0.0000	0.9612	0.0388	0.0000
1	0.0000	0.9612	0.0388	0.0000
2	0.0000	0.9612	0.0388	0.0000
3	0.0000	0.9612	0.0388	0.0000
4	0.0000	1.0000	0.0000	0.0000
5	0.0000	1.0000	0.0000	0.0000
6	0.0000	1.0000	0.0000	0.0000
7	0.0000	1.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	1.0000
9	0.0000	0.0000	0.0000	1.0000
10	0.0000	1.0000	0.0000	0.0000
11	0.0000	0.8763	0.1237	0.0000
12	0.0000	0.0002	0.0000	0.9998
13	0.0000	0.0055	0.9945	0.0000
14	0.0000	1.0000	0.0000	0.0000
15	0.0000	0.0000	1.0000	0.0000
16	0.0000	0.0055	0.9945	0.0000

A Tabela 4.1.5 apresenta as probabilidades calculadas de cada classe ser verdadeira para cada candidato. Onde:

- Classe(0): representa Classe(?), ou seja, Indefinido;
- Classe(1): representa Classe(Reprovado);
- Classe(2): representa Classe(Aprovado);
- Classe(3): representa Classe(Reavaliacao).

5. A partir das probabilidades calculadas, pode-se calcular os fatores evidenciais MC e MD (crença e descrença), respectivamente para cada Candidato. No caso do Candidato0, para a clase Reprovado os fatores de crença e descrença são os seguintes:

$$MC[Reprovado, 0] = \begin{cases} 1 & \text{se p(Reprovado)} = 1 \\ \frac{max[p(Reprovado|0), p(Reprovado)] - p(Reprovado)}{1 - p(Reprovado)} & \textit{caso contrário} \end{cases} \quad (4.11)$$

$$MD[Reprovado, 0] = \begin{cases} 1 & \text{se p(Reprovado)} = 0 \\ \frac{min[p(Reprovado|0), p(Reprovado)] - p(Reprovado)}{-p(Reprovado)} & \textit{caso contrário} \end{cases} \quad (4.12)$$

Substituindo-se pelos valores tem-se:

$$\begin{aligned} MC &= \frac{max[p(Reprovado|Candidato0), p(Reprovado)] - p(Reprovado)}{1 - p(Reprovado)} \\ MC &= \frac{max[0.9612, 0.6471] - 0.6471}{1 - 0.6471} \\ MC &= \frac{0.9612 - 0.6471}{0.3529} \\ MC &= 0.8901 \end{aligned} \quad (4.13)$$

$$\begin{aligned} MD &= \frac{min[p(Reprovado|Candidato0), p(Reprovado)] - p(Reprovado)}{-p(Reprovado)} \\ MD &= \frac{min[0.9612, 0.6471] - 0.6471}{-0.6471} \\ MD &= \frac{0.6471 - 0.6471}{-0.6471} \\ MD &= 0.0000 \end{aligned} \quad (4.14)$$

Os fatores de crença e descrença para cada classe em cada caso estão na Tabela 4.1.6. Onde:

- Classe(0): representa Classe(?);
- Classe(1): representa Classe(Reprovado);
- Classe(2): representa Classe(Aprovado);
- Classe(3): representa Classe(Reavaliacao).

Tabela 4.1.6: Fatores de Crença e Descrença para Cada Caso em Cada Classe.

Número	**Classe 0**	**Classe 1**	**Classe 2**	**Classe 3**
Candidato	**MC , MD**	**MC , MD**	**MC , MD**	**MC , MD**
0	0.0000 , 1.0000	0.8901 , 0.0000	0.0000 , 0.7802	0.0000 , 1.0000
1	0.0000 , 1.0000	0.8901 , 0.0000	0.0000 , 0.7802	0.0000 , 1.0000
2	0.0000 , 1.0000	0.8901 , 0.0000	0.0000 , 0.7802	0.0000 , 1.0000
3	0.0000 , 1.0000	0.8901 , 0.0000	0.0000 , 0.7802	0.0000 , 1.0000
4	0.0000 , 1.0000	1.0000 , 0.0000	0.0000 , 1.0000	0.0000 , 1.0000
5	0.0000 , 1.0000	1.0000 , 0.0000	0.0000 , 1.0000	0.0000 , 1.0000
6	0.0000 , 1.0000	1.0000 , 0.0000	0.0000 , 1.0000	0.0000 , 1.0000
7	0.0000 , 1.0000	1.0000 , 0.0000	0.0000 , 1.0000	0.0000 , 1.0000
8	0.0000 , 1.0000	0.0000 , 1.0000	0.0000 , 1.0000	1.0000 , 0.0000
9	0.0000 , 1.0000	0.0000 , 1.0000	0.0000 , 1.0000	1.0000 , 0.0000
10	0.0000 , 1.0000	1.0000 , 0.0000	0.0000 , 1.0000	0.0000 , 1.0000
11	0.0000 , 1.0000	0.6494 , 0.0000	0.0000 , 0.2988	0.0000 , 1.0000
12	0.0000 , 1.0000	0.0000 , 0.9997	0.0000 , 1.0000	0.9998 , 0.0000
13	0.0000 , 1.0000	0.0000 , 0.0015	0.9933 , 0.0000	0.0000 , 1.0000
14	0.0000 , 1.0000	1.0000 , 0.0000	0.0000 , 1.0000	0.0000 , 1.0000
15	0.0000 , 1.0000	0.0000 , 1.0000	1.0000 , 0.0000	0.0000 , 1.0000
16	0.0000 , 1.0000	0.0000 , 0.9915	0.9933 , 0.0000	0.0000 , 1.0000

Tabela 4.1.7: Valores de Atributos e Respectivas Distâncias

	CA	**CP**	**Experiencia**	**Teste**	**Entrevista**
Candidato0(Teste)	A	A	0	2	Regular
Candidato0(Treinamento)	F	F	0	0	Regular
`Distância_entre_Atributos`	1	1	0	1	0

4.1.2 Cálculo da Distância

Nesta seção apresenta-se o cálculo das distâncias entre os casos, de maneira a permitir a escolha dos vizinhos mais próximos de um dado caso, para obter os votos e vereditos durante a aplicação do algoritmo classificador de K-Vizinhos.

Tomando-se como exemplo o Candidato0 da Base de Teste Candidato descrita na Tabela 4.1.2, quando comparado com o Candidato0 da Base de Treinamento, tem-se a sequência de valores de atributos e respectivas distâncias indicadas na Tabela 4.1.7.

A distância entre os casos é calculada pela somatória das distâncias entre atributos. Onde:

$$\begin{aligned} &\texttt{distância}(\text{Candidato0(teste)}), (\text{Candidato0(treinamento)}) = \\ &\sum \texttt{Distância\_entre\_Atributos} \\ &= 1 + 1 + 0 + 1 + 0 \\ &= 3 \end{aligned} \tag{4.15}$$

As demais distâncias entre os candidatos da base de testes e os candidatos da base de treinamento são apresentados na Tabela 4.1.8. Onde:

- BTE significa Base de Testes;
- BTR significa Base de Treinamento;
- C(K) é o caso de número K.

Tabela 4.1.8: Distância entre os Casos da Base de Testes e os Demais da Base de Treinamento

BTR	**BTE**	**BTE**	**BTE**	**BTE**	**BTE**	**BTE**	**BTE**	**BTE**
C(K)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	3	5	5	5	5	5	3	3
1	3	5	5	5	5	5	3	3
2	3	5	5	5	5	5	3	3
3	3	5	5	5	5	5	3	3
4	4	5	5	5	5	5	0	3
5	4	5	5	5	5	5	0	3
6	4	5	5	5	5	5	0	3
7	4	5	5	5	5	5	0	3
8	5	2	5	5	3	1	5	4
9	5	1	4	5	3	2	5	4
10	4	3	3	4	2	5	4	4
11	3	4	4	5	4	4	4	4
12	5	0	4	5	3	1	5	3
13	4	3	3	3	1	3	5	4
14	3	3	5	4	5	4	3	5
15	2	5	2	0	4	5	5	5
16	2	4	3	1	3	4	5	4

Tabela 4.1.9: Cinco Vizinhos Mais Próximos na Base de Treinamento do Candidato3 da Base de Testes

Casos	**distância**
15	0
16	1
13	3
10	4
14	4

Tabela 4.1.10: Parâmetros para K=5

Caso	**Classe**	**MC**	**MD**	**distância**
Caso15	Aprovado	1.00	0.00	0.00
Caso16	Reprovado	0.00	0.99	1.00
Caso13	Aprovado	0.99	0.00	3.00
Caso10	Reprovado	1.00	0.00	4.00
Caso14	Reprovado	1.00	0.00	4.00

4.1.3 Utilizando o Classificador K-Vizinhos

Nesta seção apresenta-se um exemplo de classificação, aplicando-se os fatores evidenciais e/ou as distâncias em cinco fórmulas de votação do veredito de classe, dado pelos K-Vizinhos mais próximos, conforme apresentadas na Seção 3.7.2.

Tomando-se como exemplo o Candidato3 da base de teste Candidato, descrita na Tabela 4.1.2, com classificação dada pela votação dos cinco vizinhos mais próximos (K=5).

De acordo com a Tabela 4.1.8 os cinco vizinhos mais próximos (menor distância) na Base de Treinamento do Candidato3 da Base de Testes são, pela ordem, os casos apresentados na Tabela 4.1.9.

No exemplo escolhido, cada um dos cinco casos da base de treinamento vota em sua própria classe, com os seguintes fatores evidenciais e com as seguintes distâncias em relação ao Caso3 da Base de Testes:

1. Assim sendo, de acordo com a Fórmula 1, o peso do voto para cada um dos cinco vizinhos mais próximos do caso3 será:

$$\texttt{peso\_do\_voto}(Candidato15, Aprovado) = MC(Candidato15, Aprovado) = 1.00 \tag{4.16}$$

$$\texttt{peso\_do\_voto}(Candidato16, Reprovado) = MC(Candidato16, Reprovado) = 0.00 \tag{4.17}$$

$$\texttt{peso\_do\_voto}(Candidato13, Aprovado) = MC(Candidato13, Aprovado) = 0.99 \tag{4.18}$$

$$\texttt{peso\_do\_voto}(Candidato10, Reprovado) = MC(Candidato10, Reprovado) = 1.00 \tag{4.19}$$

$$\texttt{peso\_do\_voto}(Candidato14, Reprovado) = MC(Candidato14, Reprovado) = 1.00 \tag{4.20}$$

O veredito (classe calculada), utilizando a Fórmula 1, será:

Soma de votos em Aprovado = 1.00 + 0.99 = 1.99

Soma de votos em Reprovado = 0.00 + 1.00 + 1.00 = 2.00

Logo, a classe calculada pela Fórmula 1 é: Reprovado.

2. De acordo com a fórmula 2, o peso do voto para cada um dos cinco vizinhos mais próximos do caso3 será:

$$\texttt{peso\_do\_voto}(Candidato15, Aprovado) = \frac{1}{distancia(Candidato3, Candidato15)} = \frac{1}{0.0001} = 10000.00 \tag{4.21}$$

Neste caso, para evitar o erro causado pela divisão por zero, foi arbitrado no algoritmo um valor de 0.0001 para valor zero em qualquer das fórmulas sujeitas à divisão por zero.

$$\texttt{peso\_do\_voto}(Candidato16, Reprovado) = \frac{1}{distancia(Candidato3, Candidato16)} = \frac{1}{1.00} = 1.00 \tag{4.22}$$

$$\texttt{peso\_do\_voto}(Candidato13, Aprovado) = \frac{1}{distancia(Candidato3, Candidato13)} = \frac{1}{3.00} = 0.33 \tag{4.23}$$

$$\texttt{peso\_do\_voto}(Candidato10, Reprovado) = \frac{1}{distancia(Candidato3, Candidato10)} = \frac{1}{4.00} = 0.25 \tag{4.24}$$

$$\texttt{peso\_do\_voto}(Candidato14, Reprovado) = \frac{1}{distancia(Candidato3, Candidato14)} = \frac{1}{4.00} = 0.25 \tag{4.25}$$

O veredito (classe calculada), utilizando a fórmula 2, será:

Soma de votos em Aprovado = 10000.00 + 0.33 = 10000.33

Soma de votos em Reprovado = 1.00 + 0.25 + 0.25 = 1.50

Logo, a classe calculada pela fórmula 2 é: Aprovado.

3. De acordo com a fórmula 3, o peso do voto para cada um dos cinco vizinhos mais próximos do caso3 será:

$$\begin{aligned} &\texttt{peso\_do\_voto}(Candidato15, Aprovado) = \\ &MC(Candidato15, Aprovado) * (1 - MD(Candidato15, Aprovado)) \\ &1.00 * (1 - 0.00) = 1.00 \end{aligned} \tag{4.26}$$

$$\texttt{peso\_do\_voto}(Candidato16, Reprovado) = \\ MC(Candidato16, Reprovado) * (1 - MD(Candidato16, Reprovado)) \\ 0.00 * (1 - 0.99) = 0.00 \tag{4.27}$$

$$\texttt{peso\_do\_voto}(Candidato13, Aprovado) = \\ MC(Candidato13, Aprovado) * (1 - MD(Candidato13, Aprovado)) \\ 0.99 * (1 - 0.00) = 0.99 \tag{4.28}$$

$$\texttt{peso\_do\_voto}(Candidato10, Reprovado) = \\ MC(Candidato10, Reprovado) * (1 - MD(Candidato10, Reprovado)) \\ 1.00 * (1 - 0.00) = 1.00 \tag{4.29}$$

$$\texttt{peso\_do\_voto}(Candidato14, Reprovado) = \\ MC(Candidato14, Reprovado) * (1 - MD(Candidato14, Reprovado)) \\ 1.00 * (1 - 0.00) = 1.00 \tag{4.30}$$

O veredito (classe calculada), utilizando a fórmula 3, será:

Soma de votos em Aprovado = 1.00 + 0.99 = 1.99

Soma de votos em Reprovado = 0.00 + 1.00 + 1.00 = 2.00

Logo, a classe calculada pela fórmula 3 é: Reprovado.

4. De acordo com a fórmula 4, o peso do voto para cada um dos cinco vizinhos mais próximos do caso3 será:

$$\texttt{peso\_do\_voto}(Candidato15, Aprovado) = \frac{1}{(1-MC(Candidato15,Aprovado))*MD(Candidato15,Aprovado)*distancia(Candidato3,Candidato15)} \quad (4.31)$$

$$\frac{1}{(1-1.00)*1.00*0.00} = 10000.00$$

$$\texttt{peso\_do\_voto}(Candidato16, Reprovado) = \frac{1}{(1-MC(Candidato16,Reprovado))*MD(Candidato16,Reprovado)*distancia(Candidato3,Candidato16)} \quad (4.32)$$

$$\frac{1}{(1-0.00)*0.99*1.00} = 1.01$$

$$\texttt{peso\_do\_voto}(Candidato13, Aprovado) = \frac{1}{(1-MC(Candidato13,Aprovado))*MD(Candidato13,Aprovado)*distancia(Candidato3,Candidato13)} \quad (4.33)$$

$$\frac{1}{(1-0.99)*0.00*3.00} = 10000.00$$

$$\texttt{peso\_do\_voto}(Candidato10, Reprovado) = \frac{1}{(1-MC(Candidato10,Reprovado))*MD(Candidato10,Reprovado)*distancia(Candidato3,Candidato10)} \quad (4.34)$$

$$\frac{1}{(1-1.00)*0.00*4.00} = 10000.00$$

$$\texttt{peso\_do\_voto}(Candidato14, Reprovado) = \frac{1}{(1-MC(Candidato14,Reprovado))*MD(Candidato14,Reprovado)*distancia(Candidato3,Candidato14)} \quad (4.35)$$

$$\frac{1}{(1-1.00)*0.00*4.00} = 10000.00$$

O veredito (classe calculada), utilizando a fórmula 4, será:

Soma de votos em Aprovado = 10000.00 + 10000.00 = 20000.00

Soma de votos em Reprovado = 1.01 + 10000.00 + 10000.00 = 20001.01

Logo, a classe calculada pela fórmula 4 é: Reprovado.

5. De acordo com a fórmula 5, o peso do voto para cada um dos cinco vizinhos mais

próximos do caso3 será:

$$\begin{aligned}&\texttt{peso\_do\_voto}(Candidato15, Aprovado) = \\ &MC(Candidato15, Aprovado) * (1 - MD(Candidato15, Aprovado))* \\ &\frac{1}{distancia(Candidato3,Candidato15)} \\ &1.00 * (1 - 0.00) * \frac{1}{0.00001} = 10000.00\end{aligned} \tag{4.36}$$

$$\begin{aligned}&\texttt{peso\_do\_voto}(Candidato16, Reprovado) = \\ &MC(Candidato16, Reprovado) * (1 - MD(Candidato16, Reprovado))* \\ &\frac{1}{distancia(Candidato3,Candidato16)} \\ &0.00 * (1 - 0.99) * \frac{1}{1.00} = 0.00\end{aligned} \tag{4.37}$$

$$\begin{aligned}&\texttt{peso\_do\_voto}(Candidato13, Aprovado) = \\ &MC(Candidato13, Aprovado) * (1 - MD(Candidato13, Aprovado))* \\ &\frac{1}{distancia(Candidato3,Candidato13)} \\ &0.99 * (1 - 0.00) * \frac{1}{3.00} = 0.33\end{aligned} \tag{4.38}$$

$$\begin{aligned}&\texttt{peso\_do\_voto}(Candidato10, Reprovado) = \\ &MC(Candidato10, Reprovado) * (1 - MD(Candidato10, Reprovado))* \\ &\frac{1}{distancia(Candidato3,Candidato10)} \\ &1.00 * (1 - 0.00) * \frac{1}{4.00} = 0.25\end{aligned} \tag{4.39}$$

$$\begin{aligned}&\texttt{peso\_do\_voto}(Candidato14, Reprovado) = \\ &MC(Candidato14, Reprovado) * (1 - MD(Candidato10, Reprovado))* \\ &\frac{1}{distancia(Candidato3,Candidato14)} \\ &1.00 * (1 - 0.00) * \frac{1}{4.00} = 0.25\end{aligned} \tag{4.40}$$

Tabela 4.1.11: Resultados do Classificador K=5 para o Candidato3 da Base de Teste

Fórmula	Descrição	Veredito	CCD
1	`MC`	Reprovado	Inconsistente (Erro)
2	`1/distancia`	Aprovado	Consistente (Acerto)
3	`MC*(1-MD)`	Reprovado	Inconsistente (Erro)
4	`1/{(1-MC)*MD*distancia)}`	Reprovado	Inconsistente (Erro)
5	`MC*(1-MD)*1/distancia`	Aprovado	Consistente (Acerto)

O veredito (classe calculada), utilizando a fórmula 5, será:

Soma de votos em Aprovado = 10000.00 + 0.33 = 10000.33

Soma de votos em Reprovado = 0.00 + 0.25 + 0.25 = 0.50

Logo, a classe calculada pela fórmula 5 é: Aprovado.

Os resultados do classificador para cada fórmula estão resumidos na Tabela 4.1.11. Onde:

- CCD: representa a Consistência da Classe Declarada na entrada do novo caso na Base de Teste.

As seguintes considerações são importantes com relação aos resultados das cinco fórmulas:

- Evidentemente a base Candidato é fictícia, muito pequena e possui inconsistências propositais;
- O caso escolhido como exemplo (Candidato3) é consistente em relação à sua classe declarada (Aprovado);
- Neste exemplo, as fórmulas que levaram em conta a distância tiveram vereditos corretos, com exceção da fórmula 4. Neste caso, a utilização do inverso dos graus de crença e descrença conferiu um peso excessivo ao voto dos vizinhos com grau de crença 1 (um) ou grau de descrença 0 (zero), levando a um resultado incorreto;
- A fórmula 5 parece ser a mais indicada, por utilizar todos os fatores evidenciais como também a distância, porém sem conferir peso excessivo aos graus de crença e descrença.

4.1.4 Cálculo da Precisão e do Recobrimento do Algoritmo K-Vizinhos

Uma vez obtidos os resultados de classificação dos casos para cada fórmula de votação, é necessário avaliar a qualidade de detecção de inconsistências do algoritmo classificador K-Vizinhos. Para isso, utiliza-se o cálculo da precisão e do recobrimento de cada fórmula de votação, de acordo com as seguintes definições:

- P: representa a Precisão;
- R: representa a Recobrimento;
- MP: representa a Média da Precisão;
- MR: representa a Média do Recobrimento.

1. **Precisão**: representa a qualidade do algoritmo em acertar a classificação correta, em relação ao total de casos atribuídos pelo algoritmo àquela classe (classe calculada).

$$P = \frac{Número\ de\ Classificações\ Corretas\ para\ a\ Classe\ C}{Número\ Total\ de\ Casos\ Classificados} \tag{4.41}$$

2. **Recobrimento**: representa a qualidade do algoritmo em acertar a classificação correta, em relação ao total de casos declarados na base com aquela classe (classe declarada). É calculado dividindo o número de classificações corretas para uma determinada classe C pela quantidade de casos com aquela classificação C. Ou seja, o recobrimento avalia, para cada classe, o quão bem elas foram cobertas em termos de classificação.

$$R = \frac{Número\ de\ Classificações\ Corretas\ para\ uma\ Determinada\ Classe\ C}{Quantidade\ de\ Casos\ com\ aquela\ Classificação\ C} \tag{4.42}$$

Os resultados apresentados pelo algoritmo são na verdade as médias da Precisão e Recobrimento de todas as classes:

$$MP = \frac{\sum Precisão\ de\ cada\ Classe}{Número\ Total\ de\ Classes} \tag{4.43}$$

$$MR = \frac{\sum Recobrimento\ de\ cada\ Classe}{Número\ Total\ de\ Classes} \tag{4.44}$$

Tabela 4.1.12: Base de Testes Candidato para K=5 e Fórmula 1

Número Caso	**Classe Real**	**Classe Atribuída**	**Resultado**
0	2.00	1.00	Erro
1	3.00	3.00	Acerto
2	2.00	2.00	Acerto
3	2.00	1.00	Erro
4	2.00	3.00	Erro
5	3.00	3.00	Acerto
6	1.00	1.00	Acerto
7	3.00	1.00	Erro

Tabela 4.1.13: Base de Testes Candidato para K=5 e Fórmula 2

Número Caso	**Classe Real**	**Classe Atribuída**	**Resultado**
0	2.00	1.00	Erro
1	3.00	3.00	Acerto
2	2.00	2.00	Acerto
3	2.00	2.00	Acerto
4	2.00	2.00	Acerto
5	3.00	3.00	Acerto
6	1.00	1.00	Acerto
7	3.00	1.00	Erro

Tomando-se como exemplo a Base de Teste Candidato, para os cinco vizinhos mais próximo (K=5) e aplicando-se as cinco fórmulas de votação tem-se os seguintes cálculos dos índices de precisão e recobrimento apresentados nas Tabelas 4.1.12 a 4.1.16.

Tabela 4.1.14: Base de Testes Candidato para K=5 e Fórmula 3

Número Caso	**Classe Real**	**Classe Atribuída**	**Resultado**
0	2.00	1.00	Erro
1	3.00	3.00	Acerto
2	2.00	2.00	Acerto
3	2.00	1.00	Erro
4	2.00	3.00	Erro
5	3.00	3.00	Acerto
6	1.00	1.00	Acerto
7	3.00	1.00	Erro

Tabela 4.1.15: Base de Testes Candidato para K=5 e Fórmula 4

Número Caso	**Classe Real**	**Classe Atribuída**	**Resultado**
0	2.00	1.00	Erro
1	3.00	3.00	Acerto
2	2.00	2.00	Acerto
3	2.00	1.00	Erro
4	2.00	2.00	Acerto
5	3.00	3.00	Acerto
6	1.00	1.00	Acerto
7	3.00	2.00	Erro

Tabela 4.1.16: Base de Testes Candidato para K=5 e Fórmula 5

Número Caso	**Classe Real**	**Classe Atribuída**	**Resultado**
0	2.00	1.00	Erro
1	3.00	3.00	Acerto
2	2.00	2.00	Acerto
3	2.00	2.00	Acerto
4	2.00	3.00	Erro
5	3.00	3.00	Acerto
6	1.00	1.00	Acerto
7	3.00	1.00	Erro

Tabela 4.1.17: Base de Testes Candidato Acertos por Classe Fórmula 1

Classe	Acertos	Erros	Índice de Acerto	Precisão	Recobrimento
0	0	0	1.00	1.00	1.00
1	1	0	1.00	0.25	1.00
2	1	3	0.25	1.00	0.25
3	2	1	0.67	0.67	0.67

Tabela 4.1.18: Base de Testes Candidato Acertos por Classe Fórmula 2

Classe	Acertos	Erros	Índice de Acerto	Precisão	Recobrimento
0	0	0	1.00	1.00	1.00
1	1	0	1.00	0.33	1.00
2	3	1	0.75	1.00	0.75
3	2	1	0.67	1.00	0.67

O cálculo dos acertos por classe utilizando as cinco fórmulas de votação são apresentados nas tabelas 4.1.17 até 4.1.21.

Para o cálculo da média da precisão e do recobrimento faz-se a somatória dos índices individuais de cada classe e divide pelo número total de classes. Como nos exemplos abaixo, para a Fórmula 1:

$$\begin{aligned} \texttt{MP} &= \frac{\sum \text{Precisão de Cada Classe}}{\text{Número Total de Classes}} \\ \texttt{MP} &= \frac{1.00+0.25+1.00+0.67}{4} \\ \texttt{MP} &= \frac{2.92}{4} = 0.7292 \end{aligned} \tag{4.45}$$

Tabela 4.1.19: Base de Testes Candidato Acertos por Classe Fórmula 3

Classe	Acertos	Erros	Índice de Acerto	Precisão	Recobrimento
0	0	0	1.00	1.00	1.00
1	1	0	1.00	0.25	1.00
2	1	3	0.25	1.00	0.25
3	2	1	0.67	0.67	0.67

Tabela 4.1.20: Base de Testes Candidato Acertos por Classe Fórmula 4

Classe	Acertos	Erros	Índice de Acerto	Precisão	Recobrimento
0	0	0	1.00	1.00	1.00
1	1	0	1.00	0.33	1.00
2	2	2	0.50	0.67	0.50
3	2	1	0.67	1.00	0.67

Tabela 4.1.21: Base de Testes Candidato Acertos por Classe Fórmula 5

Classe	Acertos	Erros	Índice de Acerto	Precisão	Recobrimento
0	0	0	1.00	1.00	1.00
1	1	0	1.00	0.33	1.00
2	2	2	0.50	1.00	0.50
3	2	1	0.67	0.67	0.67

$$\begin{aligned} \texttt{MR} &= \frac{\sum \text{Recobrimento de Cada Classe}}{\text{Número Total de Classes}} \\ \texttt{MR} &= \frac{1.00+1.00+0.25+0.67}{4} \\ \texttt{MR} &= \frac{2.92}{4} = 0.7292 \end{aligned} \tag{4.46}$$

4.2 Principais Características das Bases Utilizadas

Para os testes, foram utilizadas dez bases: Candidato (criada somente para exemplo) e suas características já foram descritas na Seção 4.1.

As outras nove bases de casos Annealing (Têmpera), Wisconsin Breast Cancer (Câncer de Mama de Wisconsin), Dermatology (Dermatologia), Chess - King+Rook x King+Pawn on a 7 cuja abreviatura usual é KRKPA7 (Xadrez - Torre do Rei x Peão do Rei), Wine Recognition (Reconhecimento de Vinho), Tic-Tac-Toe Endgame (Finaliza Jogo-da-Velha), Iris (Íris), Zoo (Zoológico) e Haberman's Survival (Sobrevivência de Haberman) foram disponibilizadas pelo UCI [11] (Repository of Machine Learning Databases), da Universidade da Califórnia. As principais características que cada uma delas possui são apresentadas nesta seção. As demais características e também o uso passado das mesmas são apresentadas no *Apêndice A* deste trabalho.

A Tabela 4.2.22 apresenta um quadro resumo das principais características das bases utilizadas para os testes. Onde:

- Base: representa o nome da base;
- Tamanho: representa o número de instâncias;
- Classes: representa o número de classes que possui;
- Contínuo: representa o número de atributos de valores contínuos;
- Discreto: representa o número de atributos de valores discretos;
- Faltantes: indica a existência ou não de atributos com valores faltantes.

Tabela 4.2.22: Principais Características das Bases Utilizadas

Base	**Tamanho**	**Classes**	**Contínuos**	**Discretos**	**Faltantes**
Dermatologia	366	6	1	33	sim
Cancer de Mama	699	2	0	10	sim
Vinho	178	3	13	0	não
Têmpera	798	6	6	32	sim
Íris	150	3	0	0	não
Xadrez	3196	2	0	36	não
Zoológico	101	7	0	0	não
Jogo-da-Velha	958	2	0	9	não
Haberman	306	1	0	0	não

Capítulo 5

Resultados dos Testes

5.1 Tabelas de Resultados dos Testes

Ao aplicar-se as cinco diferentes fórmulas de votação apresentadas na Seção 3.7.2 sobre as nove bases descritas na Seção 4.2, foram obtidos os resultados apresentados nas tabelas 5.1.1 a 5.1.9, onde:

- MP: representa a Média da Precisão;
- MR: representa a Média do Recobrimento;
- K: representa a quantidade de vizinhos para um K específico e uma fórmula específica.

Tabela 5.1.1: Resultados dos Testes na Base Têmpera

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.9907,0.9621	0.9907,0.9621	0.9907,0.9621	0.9907,0.9621	0.9907,0.9621	0.9907,0.9621
3	0.9974,0.9630	0.9991,0.9848	0.9974,0.9630	0.9898,0.9205	0.9991,0.9848	0.9974,0.0030
5	0.9907,0.9621	0.9974,0.9280	0.9907,0.9621	0.9898,0.9205	0.9982,0.9697	0.9881,0.8371
7	0.9948,0.8380	0.9956,0.8447	0.9948,0.8380	0.9898,0.8205	0.9848,0.8380	0.9948,0.8380

Tabela 5.1.2: Resultados dos Testes na Base Câncer de Mama

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.9607,0.9626	0.9607,0.9626	0.9607,0.9626	0.9607,0.9626	0.9607,0.9626	0.9607,0.9626
3	0.9624,0.9604	0.9624,0.9604	0.9624,0.9604	0.9607,0.9626	0.9624,0.9604	0.9624,0.9604
5	0.9690,0.9607	0.9690,0.9607	0.9690,0.9607	0.9607,0.9626	0.9690,0.9607	0.9690,0.9607
7	0.9665,0.9562	0.9665,0.9562	0.9665,0.9562	0.9607,0.9626	0.9665,0.9562	0.9665,0.9562

Tabela 5.1.3: Resultados dos Testes na Base Dermatologia

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.9573,0.9578	0.9573,0.9578	0.9573,0.9578	0.9573,0.9578	0.9573,0.9578	0.9573,0.9578
3	0.9661,0.9653	0.9661,0.9653	0.9661,0.9653	0.9573,0.9578	0.9661,0.9653	0.9661,0.9653
5	0.9661,0.9653	0.9661,0.9653	0.9661,0.9653	0.9573,0.9578	0.9661,0.9653	0.9610,0.9573
7	0.9721,0.9721	0.9721,0.9721	0.9721,0.9721	0.9573,0.9578	0.9721,0.9721	0.9721,0.9721

Tabela 5.1.4: Resultados dos Testes na Base Xadrez

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.9371,0.9372	0.9371,0.9372	0.9371,0.9372	0.9371,0.9372	0.9371,0.9372	0.9371,0.9372
3	0.9161,0.9148	0.9651,0.9651	0.9161,0.9148	0.9225,0.9225	0.9203,0.9190	0.9630,0.9629
5	0.9064,0.9049	0.9675,0.9668	0.9064,0.9049	0.8952,0.8958	0.9147,0.9134	0.9637,0.9624
7	0.9029,0.9014	0.9665,0.9652	0.9029,0.9014	0.8737,0.8745	0.9115,0.9097	0.9613,0.9605

Tabela 5.1.5: Resultados dos Testes na Base Reconhecimento de Vinho

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.9583,0.9392	0.9583,0.9392	0.9583,0.9392	0.9583,0.9392	0.9583,0.9392	0.9583,0.9392
3	0.9520,0.9424	0.9520,0.9424	0.9520,0.9424	0.9583,0.9392	0.9520,0.9424	0.9520,0.9424
5	0.9623,0.9580	0.9623,0.9580	0.9623,0.9580	0.9583,0.9392	0.9623,0.9580	0.9623,0.9580
7	0.9623,0.9580	0.9623,0.9580	0.9623,0.9580	0.9583,0.9392	0.9623,0.9580	0.9623,0.9580

Tabela 5.1.6: Resultados dos Testes na Base Jogo-da-Velha

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.8531,0.8406	0.8531,0.8406	0.8531,0.8406	0.8531,0.8406	0.8531,0.8406	0.8531,0.8406
3	0.9025,0.8937	0.9323,0.9111	0.9025,0.8937	0.8346,0.8201	0.9025,0.8937	0.9323,0.9111
5	0.8673,0.8610	0.9516,0.9502	0.8673,0.8610	0.7923,0.7822	0.8673,0.8610	0.9516,0.9502
7	0.8301,0.8249	0.9739,0.9692	0.8301,0.8249	0.7622,0.7564	0.8301,0.8249	0.9739,0.9692

Tabela 5.1.7: Resultados dos Testes na Base Íris

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.9196,0.9219	0.9196,0.9219	0.9196,0.9219	0.9196,0.9219	0.9196,0.9219	0.9196,0.9219
3	0.9412,0.9412	0.9306,0.9265	0.9412,0.9412	0.9196,0.9219	0.9412,0.9412	0.9306,0.9265
5	0.9306,0.9265	0.9306,0.9265	0.9306,0.9265	0.9196,0.9219	0.9306,0.9265	0.9211,0.9118
7	0.9206,0.9265	0.9206,0.9265	0.9206,0.9265	0.9072,0.9072	0.9412,0.9412	0.9306,0.9265

Tabela 5.1.8: Resultados dos Testes na Base Zoológico

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.8988,0.9083	0.8988,0.9083	0.8988,0.9083	0.8988,0.9083	0.8988,0.9083	0.8988,0.9083
3	0.7738,0.8250	0.7738,0.8250	0.7738,0.8250	0.8988,0.9083	0.7738,0.8250	0.7738,0.8250
5	0.7738,0.8250	0.7738,0.8250	0.7738,0.8250	0.8988,0.9083	0.7738,0.8250	0.7738,0.8250
7	0.7321,0.8250	0.7321,0.8250	0.7321,0.8250	0.8988,0.9083	0.7321,0.8250	0.7321,0.8250

Tabela 5.1.9: Resultados dos Testes na Base Sobrevivência de Habermann

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.6852,0.6767	0.6852,0.6767	0.6852,0.6767	0.6852,0.6767	0.6852,0.6767	0.6852,0.6767
3	0.6654,0.6663	0.7474,0.6986	0.6654,0.6663	0.6121,0.6334	0.6997,0.6771	0.7594,0.6880
5)	0.5506,0.6503	0.8106,0.6935	0.5506,0.6503	0.6406,0.6436	0.7594,0.6880	0.7803,0.6827
7	0.5543,0.6667	0.9106,0.6935	0.5543,0.6667	0.6319,0.6325	0.7222,0.6720	0.5543,0.6767

Tabela 5.2.10: Resultados dos Testes na Base Candidato

K	Fórmula 1	Fórmula 2	Fórmula 3	Fórmula 4	Fórmula 5	**Fórmula 6**
	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR	MP , MR
1	0.8750,0.9167	0.8750,0.9167	0.8750,0.9167	0.8750,0.9167	0.8750,0.9167	0.8750,0.9167
3	0.8333,0.8542	0.8333,0.8542	0.8333,0.8542	0.7500,0.7917	0.8750,0.9167	0.8333,0.8542
5	0.7292,0.7292	0.8333,0.8542	0.7292,0.7292	0.7500,0.7917	0.7500,0.7917	0.7292,0.7292
7	0.4375,0.6667	0.8000,0.7292	0.4375,0.6667	0.6250,0.6417	0.7500,0.7917	0.4667,0.6667

5.2 Interpretação dos Resultados

Primeiramente, quanto aos resultados para a base "Candidato"da Tabela 5.2.10 nota-se que os percentuais da média da precisão e da média do recobrimento são bastante baixos. É importante observar que, nos testes, a divisão da base resultou em 17 casos para treinamento e 8 para testes. Para uma quantidade tão pequena de casos, o algoritmo não tem casos de treinamento suficientes para estabelecer uma relação apropriada entre os atributos, portanto resultado em um desempenho ruim. Além disso esta base fictícia foi construída com inconsistências propositais, o que diminui naturalmente os índices de precisão e recobrimento.

Quanto às bases utilizadas, vale a pena deixar claro que nove bases já podem ser consideradas para tirar conclusões que, embora não definitivas, já constituem um estudo de caso interessante.

O primeiro aspecto interessante é comparar os resultados do algoritmo de uma base para outra. A Fórmula 6 representa o classificador K-Vizinhos tradicional, e foi utilizada para permitir uma comparação com as fórmulas que utilizam fatores evidenciais. Por isso, os resultados da Fórmula 6 não serão considerados no cálculo da média das médias dos índices de precisão e recobrimento.

A Tabela 5.2.11 apresenta um quadro resumo dos resultados obtidos, com uma classificação em ordem crescente dos resultados da média de todos os valores de *MP* de cada base. Similarmente a Tabela 5.2.12 apresenta um quadro resumo dos resultados obtidos, para os valores de *MR* de cada base.

A Tabela 5.2.13 apresenta um quadro resumo dos resultados obtidos, com uma classificação em ordem crescente dos melhores resultados da média das médias da precisão

Tabela 5.2.11: Resultados Obtidos nas Bases em Relação a MP

Nome da Base	**No. Instâncias**	**No. Atributos**	$\overline{\text{MP}}$
Têmpera	798	38	0.9931
Cancer de Mama	699	10	0.9638
Dermatologia	366	34	0.9637
Reconhecimento de Vinho	178	13	0.9586
Íris	150	4	0.9261
Xadrez	3196	36	0.9236
Jogo-da-Velha	958	9	0.8656
Zoológico	101	17	0.8154
Haberman	306	3	0.6750

Tabela 5.2.12: Resultados Obtidos nas Bases em Relação a MR

Nome da Base	**No. Instâncias**	**No. Atributos**	$\overline{\text{MR}}$
Dermatologia	366	34	0.9636
Cancer de Mama	699	10	0.9604
Reconhecimento de Vinho	178	13	0.9473
Têmpera	798	38	0.9274
Íris	150	4	0.9268
Xadrez	3196	36	0.9229
Zoológico	101	17	0.8583
Jogo-da-Velha	958	9	0.8565
Haberman	306	3	0.6690

para cada fórmula de votação, onde F_i representa a Fórmula i para $i = 1, 6$. A coluna **F6 Controle** apenas mostra os valores da média das médias dos índices de precisão. Similarmente a Tabela 5.2.14 apresenta um quadro resumo dos resultados obtidos, da média das médias do recobrimento para cada fórmula de votação.

As Tabelas 5.2.15 e 5.2.16 representam a diferença percentual de MP e MR de cada fórmula de votação em relação à Fórmula de Controle F6.

Tabela 5.2.13: Resultados Obtidos nas Bases em Relação a MP para cada Fórmula de Votação

Nome da Base	**Fórmula**	**Fórmula**	**Fórmula**	**Fórmula**	**F6 Controle**
Têmpera	F2 0.9957	F1 0.9934 F3 0.9934	F5 0.9932	F4 0.9900	0.9931
Dermatologia	F1 0.9654 F2 0.9654 F3 0.9654 F5 0.9654	F4 0.9573			0.96337
Cancer	F1 0.9646 F2 0.9646 F3 0.9646 F5 0.9646	F4 0.9607			0.9638
Xadrez	F2 0.9590	F5 0.9209	F1 0.9156 F3 0.9156	F4 0.9071	0.9236
Vinho	F1 0.9587 F2 0.9587 F3 0.9587 F5 0.9587	F4 0.9583			0.9586
Íris	F5 0.9331	F1 0.9280 F3 0.9280	F2 0.9253	F4 0.9165	0.9261
Jogo-da-Velha	F2 0.9277	F1 0.8632 F3 0.8632 F5 0.8632	F4 0.8105		0.8656
Zoológico	F4 0.8988	F1 0.7946 F2 0.7946 F3 0.7946 F5 0.7946			0.8152
Haberman	F2 0.7884	F5 0.7166	F4 0.6424	F1 0.6138 F3 0.6138	0.6750

Tabela 5.2.14: Resultados Obtidos nas Bases em Relação a MR para cada Fórmula de Votação

Nome da Base	**Fórmula**	**Fórmula**	**Fórmula**	**Fórmula**	**F6 Controle**
Dermatologia	F1 0.9651 F2 0.9651 F3 0.9651 F5 0.9651	F4 0.9578			0.9636
Cancer	F4 0.9626	F1 0.9599 F2 0.9599 F3 0.9599 F5 0.9599			0.9604
Xadrez	F2 0.9585	F5 0.9198	F1 0.9145 F3 0.9145	F4 0.9075	0.9229
Vinho	F1 0.9494 F2 0.9494 F3 0.9494 F5 0.9494	F4 0.9392			0.9473
Têmpera	F5 0.9386	F1 0.9313 F3 0.9313	F2 0.9299	F4 0.9059	0.9274
Íris	F5 0.9327	F1 0.9290 F3 0.9290	F2 0.9253	F4 0.9182	0.9268
Jogo-da-Velha	F2 0.9177	F1 0.8550 F3 0.8550 F5 0.8550	F4 0.7998		0.8565
Zoológico	F4 0.9083	F1 0.8458 F2 0.8458 F3 0.8458 F5 0.8458			0.8583
Haberman	F2 0.6905	F5 0.6784	F1 0.6650 F3 0.6650	F4 0.6465	0.6690

Tabela 5.2.15: Diferença Percentual de MP de Cada Fórmula em Relação à Fórmula 6

Nome da Base				
Têmpera	F2 0.30%	F1 e F3 0.07%	F5 0.05%	F4 - 0.27%
Dermatologia	F1, F2, F3 e F5 0.13%	F4 - 0.70%		
Cancer	F1, F2, F3 e F5 0%	F4 - 0.40%		
Xadrez	F2 0.29%	F5 - 3.69%	F1 e F3 - 4.24%	F4 - 5.13%
Vinho	F1, F2, F3 e F5 0%	F4 0.04%		
Íris	F5 0.82%	F1 e F3 0.27%	F2 - 0.02%	F4 0.97%
Jogo-da-Velha	F2 0%	F1, F3 e F5 - 6.95%	F4 - 12.63%	
Zoológico	F4 13.1%	F1, F2, F3 e F5 0%		
Haberman	F2 13.4%	F5 3.13%	F4 - 7.54%	F1 e F3 - 11.65%

Tabela 5.2.16: Diferença Percentual de MR de Cada Fórmula em Relação à Fórmula 6

Nome da Base				
Dermatologia	F1, F2, F3 e F5 0.20%	F4 0.55%		
Cancer	F4 0.27%	F1, F2, F3 e F5 - 0.01%		
Xadrez	F2 - 0.74%	F5 - 4.75%	F1 e F3 - 5.30%	F4 - 6.02%
Vinho	F1, F2, F3 e F5 0%	F4 - 1.07%		
Têmpera	F5 4.28%	F1 e F3 3.4%	F2 3.32%	F4 0.65%
Íris	F5 1.19%	F1 e F3 0.79%	F2 0.39%	F4 0.37%
Jogo-da-Velha	F2 - 0.01%	F1, F3 e F5 - 6.84%	F4 - 12.85%	
Zoológico	F4 7.38%	F1, F2, F3 e F5 0%		
Haberman	F2 1.76%	F5 - 0.01%	F1 e F3 1.98%	F4 - 4.71%

5.2.1 Interpretação e Considerações sobre os Resultados

Uma primeira análise revela que os resultados de precisão e recobrimento são piores para a base Sobrevivência de Haberman. Ao observar com visão de especialista os atributos da base Sobrevivência de Haberman fica clara a razão do mal desempenho nesta base. Existem apenas três atributos, sendo que o atributo ano de operação não deve apresentar uma correlação forte com a classificação. É provável que o atributo quantidade de nódulos auxiliares apresente uma correlação com o tempo de sobrevivência do paciente, e que o atributo idade do paciente também apresente uma correlação com a classificação, porém em menor índice que a quantidade de nódulos. O baixo desempenho, neste caso, indica mais a pouca correlação entre os atributos da base e a classificação do que um pequeno grau de acerto do algoritmo.

As bases Zoológico e Jogo-da-Velha apresentam índices de precisão e recobrimento intermediários (entre 80% e 90%). A base Jogo-da-Velha apresenta a posição de final de jogo tendo como classe a vitória ou não do jogador "x´´. Caso haja empate, a classe resultante será a não vitória do jogador "x´´, e assim tanto um empate quanto a vitória do jogador "o´´ terão a mesma classe. Como os atributos são posicionais, não surpreende que o melhor resultado seja o da Fórmula 2 de votação, que utiliza somente a métrica da distância. A Fórmula 4, que pode atribuir um peso do voto muito grande para graus de crença um e descrença zero, tem logicamente os piores índices de precisão e recobrimento. Levando-se em conta que a classe não ganha "x"soma os casos de vitória do jogador "o"e de empate, pode-se considerar os índices obtidos como muito bons.

A base Zoológico apresenta uma correlação entre atributos simbólicos característicos de espécies animais e sua classificação taxonômica. Levando-se em conta que existem vinte classes possíveis para dezesete atributos, pode-se considerar os resultados satisfatórios. Nesta base surpreendentemente a Fórmula 4 apresentou os melhores índices, enquanto foi a pior na grande maioria das bases. De fato, a classificação taxonômica dos animais tende a utilizar como determinante de classe um ou poucos atributos. Por exemplo: presença de penas é determinante para a classificação de um animal como ave. A secreção de leite é determinante da classe mamífero. Por isso, a Fórmula 4 que confere um peso do voto muito elevado ao graus de crença um e descrença zero, obteve resultados consideravelmente melhores que as outras quatro fórmulas de votação.

As outras bases (Xadrez, Íris, Reconhecimento de Vinho, Dermatologia, Câncer de Mama e Têmpera) apresentam índices altos de precisão e recobrimento (acima de 90%), com destaque para as bases Dermatologia, Câncer de Mama, Têmpera e Reconhecimento de Vinho. É de se destacar que, apesar do grande número de valores desconhecidos na base Têmpera, o índice de precisão (0.9931) foi muito bom.

Quanto a base Xadrez, levando em consideração que se trata de uma base posicional retratando um fim de jogo, os resultados podem ser considerados muito bons. Novamente, como no caso da base Jogo-da-Velha as melhores fórmulas foram a Fórmula 2 (que leva em consideração somente a distância) e a segunda melhor foi a Fórmula 5 (que leva em consideração a distância e os fatores de crença e descrença), e a pior fórmula foi a Fórmula 4 (que confere peso do voto muito alto a fatores de crença um ou descrença zero).

Ao observar-se as características dos atributos e classes das bases Têmpera, Reconhecimento de Vinho, Câncer de Mama e Dermatologia, pode-se concluir preliminarmente que o algoritmo tem melhor desempenho em bases com elevado número de atributos e poucas classes, como também em bases cuja contribuição para classificação é dada pelo conjunto de atributos. Por exemplo, na Base Reconhecimento de Vinho, a classificação de origem do vinho está relacionada com uma análise física e química na qual os atributos se combinam de forma complexa para compor a classificação. Na base Dermatologia, diversos atributos se combinam para o diagnóstico, que é reconhecidamente complexo, pois a sintomatologia das doenças eritemato-escamosas apresenta muitos pontos em comum. Também é possível argumentar que o algoritmo teve um bom desempenho em bases complexas, tanto em bases com elevado número de atributos simbólicos (Têmpera) quanto com elevado número de atributos contínuos (Reconhecimento de Vinho), ou seja, o algoritmo desenvolvido apresenta, nestes testes, bons resultados tanto para atributos previamente discretizados como para atributos que necessitem ser discretizados pelo algoritmo de Mitchell modificado (com Largura Mínima de Valor Discreto).

5.2.2 Relação entre Precisão, Fórmulas de Votação e Número de K-Vizinhos

Para as Fórmulas 1 e 3

Quanto a classificação dos K-Vizinhos, a Fórmula 1 (que considera apenas o fator de crença) e a Fórmula 3 (que considera um menos o fator de descrença multiplicado pelo fator de crença) apresentam resultados similares, independentemente do número de vizinhos votantes. Isso se deve a existência de muitos casos com fator de crença um, e consequente fator de descrença zero, de maneira que os pesos do voto da Fórmula 1 será um e o peso do voto da Fórmula 3 será 1(1-0) = 1, ou seja, esses pesos serão iguais e determinantes no processo de votação.

A correlação entre o número de vizinhos e os índices de precisão para as fórmulas que levam em conta somente o grau de crença e descrença (Fórmulas 1 e 3) depende do tipo de base.

Nas bases de pior desempenho (Sobrevivência de Haberman), ou posicionais de fim de jogo (Xadrez e Jogo-da-Velha), existe um tendência de diminuição do índice de precisão com o aumento do número de vizinhos. No caso da base Sobrevivência de Haberman a explicação parece estar na baixa correlação entre os valores dos atributos e o resultado da classificação, o que faz com que o algoritmo Naïve Bayes tenha pouca utilidade para representar os graus de crença e descrença. Para as bases cujos atributos representam posições de fim de jogo (Xadrez e Jogo-da-Velha) as fórmulas baseadas nos graus de crença e descrença (Fórmulas 1 e 3) apresentem resultados piores à medida em que aumenta o número de vizinhos, pois não há relação entre a probabilidade de uma posição de jogo isolada aparecer e a vitória de um determinado jogador.

Na base Zoológico o aumento de vizinhos diminui a precisão dos resultados para as fórmulas 1 e 3. Conforme anteriormente explicado, a influência dos atributos com grau de crença um e descrença zero é determinante para esta base. Se o número de vizinhos aumenta, o peso do voto destes vizinhos de grau de crença elevado é diluído, induzindo ao erro de classificação.

Nas bases Íris, Câncer de Mama e Têmpera, parece haver um número ótimo de vizinhos entre 3 e 5 para as fórmulas 1 e 3, isto é, parece que o número de vizinhos não deve

ser muito pequeno nem muito alto. É possível que alguns dos atributos tenham maior influência que outros, e determinem a classe com um grau de crença 1 (descrença zero), pesando bastante nos resultados, mas não sendo corretos, a não ser quando combinados. Deste modo é preciso haver a votação de alguns vizinhos em que estes atributos aparecem, para contribuir com o acerto da classificação. Se, no entanto o número de vizinhos é muito alto, as distâncias também passam a ser altas para os vizinhos mais distantes, porém estes terão um peso do voto muito alto, induzindo ao erro de classificação.

As bases Reconhecimento de Vinho e Dermatologia apresentam um aumento da precisão proporcional ao aumento do número de vizinhos para as fórmulas 1 e 3. Como anteriormente explicado, estas bases mais complexas dependem do conjunto de atributos para classificação, de modo que os graus de crença e descrença estão bem distribuídos e apresentam boa correlação com a influência dos atributos sobre a classificação.

Para a Fórmula 4

De maneira geral a Fórmula 4 apresentou resultados de precisão constante em relação ao número de vizinhos. Como demonstrado anteriormente, quando o grau de crença é um (ou grau de descrença é zero), o resultado da fórmula assumirá um peso do voto muito alto, que será determinante mesmo com o aumento de vizinhos votantes. isso aconteceu com as bases Zoológico, Reconhecimento de Vinho, Dermatologia e Câncer de Mama. Na base Têmpera os resultados foram constantes a partir de três vizinhos e na base Íris, o resultado constante só mudou para sete vizinhos. Excessões foram, novamente a base Sobrevivência de Haberman, Xadrez e Jogo-da-Velha. Como já foi indicado, os graus de crença e descrença tem pouca ou nenhuma relação com a classificação real dos casos nestas bases.

Para a Fórmula 2

A Fórmula 2, que não considera os fatores da LP, mas tão somente a distância entre os vizinhos, apresentou um aumento consistente dos índices com o número de vizinhos para as bases Reconhecimento de Vinho, Jogo-da-Velha, Sobrevivência de Haberman e Dermatologia. No caso da base Sobrevivência de Haberman, como os graus de crença e descrença estão pouco correlacionados com os resultados, ficou claro que a métrica

da distância, e o número de vizinhos votantes melhora os resultados. É muito evidente a diferença na qualidade dos resultados da Fórmula 2 com K=7 (MP=0.9106) quando comparados com os resultados das Fórmulas 1 e 3 também para K=7 (MP=0.5543).

Também se percebe que as bases de posições de fim de jogo (Jogo-da-Velha e Xadrez) apresentam resultados melhores com o aumento do número de vizinhos para a Fórmula 2.

No caso das bases Reconhecimento de Vinho e Dermatologia, como a classificação está relacionada com o conjunto dos atributos, a métrica da distância contribui muito para os resultados, e o aumento do número de vizinhos votantes influencia positivamente no acerto do veredito.

Evidentemente na base Zoológico a precisão diminui com o número de vizinhos, pois neste caso o fator de crença tem uma influência muito grande e a métrica da distância tende a piorar os resultados. Conforme demosntrado na subseção Interpretação Genérica dos Resultados a Fórmula 4 é considerada ideal para esta base.

Para a Fórmula 5

De modo geral, a fórmula 5, que leva em conta tanto os fatores de crença e descrença quanto a distância, porém sem dar peso excessivo de voto dos casos com fator de crença 1 (descrença zero), foi a segunda em desempenho, perdendo apenas para a fórmula 2. A Fórmula 5 foi a de maior precisão para a base Íris, empatou em primeiro lugar nas bases Câncer, Dermatologia, e Reconhecimento de Vinho, e foi a segunda melhor fórmula em geral para as bases Xadrez, Jogo-da-velha, Zoológico e Sobrevivência de Haberman. Mesmo na base Têmpera, ficou em terceiro lugar por uma margem muito pequena.

Capítulo 6

Considerações Finais

6.1 Conclusões

O tratamento automático de inconsistências em sistemas de RBC com uso de fatores evidenciais da PrLE é um problema que até agora não tinha sido abordado, fazendo deste trabalho uma iniciativa inédita na área.

Nesse trabalho os atributos de uma determinada base de casos são primeiramente discretizados utilizando uma modificação do algoritmo de Mitchell [54] sensível as mudanças de classes. Em seguida, o algoritmo Naïve-Bayes é aplicado para o cálculo dos fatores evidenciais de crença e descrença das classes de cada caso. Então, aplica-se uma métrica de distância entre os casos, para em seguida obter a classificação de um caso em teste através de um algoritmo classificador K-Vizinhos mais próximos, mediante cinco fórmulas de votação que combinam os fatores evidenciais com a métrica da distância.

Deste modo, pode ser automaticamente indicada uma provável inconsistência na classificação do caso em teste, sem utilizar um conhecimento de um especialista no processo de classificação da base original.

A métrica da distância escolhida e o algoritmo de classificação dos K-Vizinhos mais próximos, principalmente utilizando as Fórmulas 2 e 5, apresenta índices de precisão e recobrimento bastante elevados para as bases mais complexas e de grande número de instâncias (casos) e atributos.

Como era de se esperar, os casos particulares de bases que contenham atributos sem relação com a classificação são os menos indicados para tratamento automático de incon-

sistências pelo algoritmo proposto, como mostrado pelos resultados da base Sobrevivência de Haberman.

No entanto, o algoritmo mostrou-se robusto e confiável na detecção automática de inconsistências mesmo em bases de tipos de correlação atributo x classe tão diferentes quanto nas bases de fim-de-jogo e bases de diagnóstico.

Em geral as bases com bom número de atributos contribuintes para a classificação, como as bases de diagnóstico (Câncer de Mama, Dermatologia, Reconhecimento de Vinho e Têmpera) e de fim-de-jogo (Xadrez e Jogo-da-Velha) obtiveram maior índice de precisão com o uso da Fórmula 2 (isto é, sem levar em conta os fatores de crença e descrença da LP), embora a base de classificação taxonômica (Zoológico) tenha obtido melhores índices de precisão com a Fórmula 4 (pois os fatores de crença 1 são determinantes para a classificação taxonômica). A Fórmula 5 que utiliza tanto os fatores de crença e descrença quanto a distância em proporcões equilibradas é bastante segura quando não for possível saber as características da base, pois, muito embora a Fórmula 2 tenha sido a melhor no geral, em bases com forte correlação da classificação com os fatores de crença e descrença, a Fórmula 2 pode ter o pior resultado. O número ideal de vizinhos, para a maioria das bases, parece estar entre três e cinco.

A principal conclusão deste trabalho é portanto de que, em caso de não se conhecer a priori as características da base, é possível e desejável combinar a métrica da distância e o classificador K-Vizinhos com os fatores de crença e descrença da PrLE. Portanto, este trabalho mostra empiricamente que a LP pode ser usada para tratar e classificar inconsistências e melhorar a performance de classificação de um sistema RBC tradicional, de forma automática e genérica, embora o ganho percentual em relação ao classificador K-Vizinhos tradicional seja pequeno.

Através da utilização conjunta dos conceitos de AM e PrLE, foi possível o desenvolvimento de um algoritmo de extrema utilidade, uma vez que no mundo real a maioria das bases de dados contém informações imperfeitas, ou incompletas, tais como: exemplos com valores faltando para atributos, erros de amostragens, falta de atributos relevantes e inconsistências.

Esse trabalho utilizou os conceitos da LP para manipulação adequada de informações inconsistentes, aplicados ao paradigma RBC. O algoritmo foi desenvolvido para encon-

trar, classificar e tratar inconsistências na base de casos de um RBC, estabelecendo um formalismo capaz de resultar em um conjunto de casos acompanhados de seus respectivos fatores evidenciais, e da medida da distância em relação aos outros casos. Desta forma o sistema resultante é capaz de analisar um conjunto de exemplos e gerar dados que possam ser usados pelo RBC para analisar novos casos tratando automaticamente as inconsistências encontradas.

6.2 Extensões e Trabalhos Futuros

A partir do trabalho desenvolvido é possível realizar outros trabalhos, tanto no sentido de estender o sistema, como avaliar melhor seu desempenho e estudar maneiras de integrar o sistema com outras aplicações.

Do ponto de vista da eficiência do sistema, o classificador poderia ser aprimorado através da implementação de um algoritmo K-Vizinhos mais eficiente. Pode-se utilizar outros algoritmos de armazenamento na memória e baseados em árvores, como por exemplo *K-D-Trees*, uma estrutura de dados proposta por Bentley [7], que estende a ABBB - Árvore Binária de Busca Balanceada em K dimensões.

No que se refere a testes, o presente trabalho se limitou a apresentar os resultados sobre um conjunto de bases de casos. Não é possível ter uma idéia adequada sobre o desempenho do sistema sem comparar com outros algoritmos de classificação, tais como árvores de decisão, algoritmos genéticos e redes neurais, [15, 64, 65]. Ainda na questão de avaliação, valeria a pena estudar outras heurísticas dentro do próprio sistema, variando a forma de avaliar a distância e também de considerar os fatores evidenciais nas fórmulas de votação de maneira diferente.

Os altos índices de precisão obtidos em bases de testes com muitos dados são uma forte indicação de que o algoritmo consegue classificar automaticamente um novo caso com razoável certeza, e assim sinalizar uma inconsistência. Um trabalho interessante seria uma avaliação e correção de inconsistências de uma base preferivelmente de diagnóstico, por um especialista, e subsequente avaliação do algoritmo.

Considerando o uso do sistema em aplicações reais, poderia ser um desafio aproveitar os resultados do sistema, tal como a classificação e os próprios fatores de modo a determinar de maneira mais precisa e eficiente como, por exemplo um diagnóstico médico.

Referências Bibliográficas

[1] A. Aamodt e E. Plaza. Case-based reasoning: Foundational issues, methodological variations and systems approaches. *Artificial Intelligence Communications*, 7(1):39–59, 1994.

[2] S. Aeberhard, D. Coomans, e de O. Vel. The classification performance of rda. Relatório Técnico 92-01, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992.

[3] S. Aeberhard, D. Coomans, e de O. Vel. Comparison of classifiers in high dimensional settings. Relatório Técnico 92-02, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992.

[4] D. W. Aha. Incremental constructive induction: An instance-based approach. *Eighth International Workshop on Machine Learning*, páginas 117–121, Evanston, 1991. Morgan Kaufmann Publisher.

[5] B. P. Allen. Case-based reasoning: Business aplications. *Communications of the ACM*, 37(3):40–42, 1994.

[6] K. D. Althoff e S. Web. Case-based reasoning and expert system development. Springer-Verlag, editor, *Lecture Notes in AI*, volume 622, páginas 145–158, Berlim, Germany, 1992.

[7] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[8] G. Bittencourt. *Inteligência Artificial - Ferramentas e Teorias*. Editora da UFSC, 2001.

[9] H. A. Blair e V. S. Subrahmanian. Paraconsistent logic programming. *Lecture Notes in Computer Science*, number 287. 7th Conference on Fondations of Software Technology and Teorical Computer Science, 1987.

[10] H. A. Blair e V. S. Subrahmanian. Paraconsistent foundations for logic programming. *Non-Classical Logic*, 5, 2:46–73, 1988.

[11] C. L. Blake e C. J. Merz. Uci - repository of machine learning databases, 1998.

[12] S. Branskat. Knowledge aquisition from cases. Springer-Verlag, editor, *Lecture Notes in AI*, volume 622, páginas 134–145, Berlim,Germany, 1992.

[13] B. G. Buchanan e E. H. Shortliffe. *Rule-Based Expet Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley Publishing Company, USA, 1984.

[14] P. Clark e T. Niblett. The cn2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, e C. Stein. *Algoritmos - Tradução da Segunda Edição Americana*. Campus, 2002.

[16] A. Cornuéjols e L. Miclet. *Apprentissage Artificiel: Concepts et Algorithmes*. Eyrolles, Août de 2002.

[17] N. C. A. Costa, J. M. Abe, J. I. da Silva Filho, A. C. Murolo, e C. F. S. Leite. *Lógica Paraconsistente Aplicada*. Atlas, 1999.

[18] N. C. A. Costa, J.P.A. Prado, J.M. Abe, B.C. Ávila, e M. Rillo. Paralog: Um prolog paraconsistente baseado em lógica anotada. *Coleção Documentos*, number 18, São Paulo, april de 1995. Instituto de Estudos Avançados, Universidade de São Paulo.

[19] B. V. Dasarathy. Nosing around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2, No. 1:67–71, 1980.

[20] G. Demiroz, H. A. Govenir, e N. Ilter. Learning differential diagnosis of eryhemato-squamous diseases using voting feature intervals. *IEEE*, páginas 147–165, 1998.

[21] D. Dubois, F. Esteva, P. Garcia, L. Godo, R. L. Mántaras, e H. Prade. A fuzzy approach. *Fuzzy Logic in Artificial Intelligence (IJCAI Workshop)*, páginas 79–90, 1997.

[22] R. O. Duda e P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Cambridge, Massachusetts, 1973.

[23] F. Enembreck. Um sistema paraconsistente para verificação automática de assinaturas manuscritas. Dissertação de Mestrado, PUCPR - Pontifícia Universidade Católica do Paraná, Curitiba, BR, 1999.

[24] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7, Part II:179–188, 1936. also in Contributions to Mathematical Statistics - John Wiley, NY, 1950.

[25] M. R. Garey e D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., New York, EUA, 1979.

[26] G. W. Gates. The reduced nearest neighbor rule. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2, No. 1:431–433, may de 1972.

[27] W. Gates, G. Cheeseman et al's autoclass ii conceptual clustering system finds 3 classes in the data. páginas 54–64, Boston, 1988. MLC Proceedings.

[28] D. Gentner. Structure mapping - a theorical framework for analogy. *Cognitive Science*, 7:155–170, 1983.

[29] S. J. Haberman. Generalized residuals for log-linear models. páginas 104–122, Boston, 1976. 9th International Biometrics Conference.

[30] K. J. Hammond. *Case-Based Planning*. Academic Press, 1989.

[31] R. Hanson e J. Stutz. Bayesian classification theory. Relatório Técnico FIA-90-12-7-01, NASA Ames Research Center, 1990.

[32] T. R. Hinrichs. *Problem Solving in Open Worlds*. Lawrence Erlbaum Associates, 1992.

[33] R. C. Holte, L. Acker, e B. W. Porter. Concept learning and the problem of small disjuncts. Austin, Texas, 1989. IJCAI.

[34] P. Indyk e R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, páginas 604–613. ACM Press, 1998.

[35] J. L. Kolodner. Reconstrutive memory, a computer model. *Cognitive Science*, 7:281–328, 1983.

[36] J. L. Kolodner. Judging which is the best case for case-based reasoner. *Case-Based Reasoning Workshop*, 1989.

[37] J. L. Kolodner. Improving human decision making through. *AI Magazine*, 12(2):52–68, 1991.

[38] J. L. Kolodner. An introduction to case-based reasoning. *AI Magazine - Review*, 6(1):3–34, 1992.

[39] J. L. Kolodner. *Case-Based Reasoning*, volume 10, páginas 195–199. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[40] J. L. Kolodner e Leake D. *A Tutorial Introduction to Case-Based Reasoning: Experiences, Lessons, & Future Directions*. AAAI Press, The MIT Press, California, CA, October de 1996.

[41] P. Kotton. *Using Experience in Learnning and Problem Solving*. Tese de Doutorado, Massachusssets Institute of Technology - Laboratory of Computer Sciense, Massachussets, October de 1989.

[42] D. Krause. A lógica paraconsistente, 2004.

[43] J. M. Landwehr, D. Pregibon, e A. C. Shoemaker. Graphical models for assessing logistic regression models. *of the American Statistical Association*, 79:61–83, 1984.

[44] G. S. Lira e M. Fantinato. Arquitetura de um sistema cbr, 2002.

[45] W.D. Lo. *Logistic Regression Trees*. Tese de Doutorado, Department of Statistics, University of Wisconsin, 1993.

[46] C. Marcus. *Prolog Programming: Aplication for Database Systems, Expert Systems and Natural language Systems*. Addison-Wesley Publishing Company, USA, 1986.

[47] H. C. Martins, C. I. A. Costa, e G. L. Torres. *Generalization of Fuzzy and Classic Logic in NPL2v, Advances in System Science: Measurement, Circuits and Control - Eletrical and Computer Enginnering*. Lawrence Erlbaum Associates, New Jersey, 2001.

[48] C. J. Matheus. Adding domain knowledge to sbl through feature construction. *Eighth National Conference on Artificial Intelligence*, páginas 803–808, Boston, MA, 1990. AAAI Press.

[49] C. J. Matheus e L. A. Rendell. Constructive induction on decision trees. *Eleventh International Joint Conference on Artificial Intelligence*, páginas 645–650, Detroit, MI, 1989. Morgan Kaufmann Publisher.

[50] A. McCallum e K. Nigam. A comparision of event models for naïve bayes text classification. Number AAAI-98. 5th National Conference on Artificial Intelligence, 1998.

[51] R. McCartney. Case-based planning meets the frame problem. *International Conference on AI Planning Systems*, San Mateo, CA, 1992. 1º College Parck, Morgan Kaufmann Publisher.

[52] M. A. Minsky. *A Framework for Representation Knowledge*. McGraw-Hill, New York, 1975.

[53] M. A. Minsky. *The Society of Mind*. Touchstone Book, New York, 1985.

[54] T. M. Mitchell. *Machine Learning*. WCB/McGraw-Hill, 1997.

[55] S. Muggleton. *Structuring Knowledge by Asking Questions in Progress in Machine Learning*. Sigma Press, Wilmslow, GB, 1987.

[56] R. E. Neapolitan. *Probabilistic Reasoning In Expert Systems: Theory and Algorithms.* Wiley-Interscience Publication, USA, 1990.

[57] C. Owens. Integration feature extraction and memory-based learning. *Machine Learning*, 10:311–339, 1993.

[58] C. M. Papadimitriou. *Computational Complexity.* Addison-Wesley Publishing Company, Inc., New York, EUA, August de 1994.

[59] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, California, 2 ed. edition, 1988.

[60] B. Porter e R. P. Bareiss. An experiment in knowledge acquisition for heuristic classification cases. *III Proceedings of the First International Advancces in Learning (IMAL)*, páginas 159–174, Les Arcs, France, 1986.

[61] J. R. Quinlan. *C4.5: Programs for machine Learning.* Morgan Kaufmann, San Mateo, California, 1993.

[62] K. Racine e Q. Yang. On the consistency management of large case bases: the case for validation. *AAAI Technical Report - Verification and Validation Workshop*, Burnaby, Canada, 1996.

[63] K. Racine e Q. Yang. Redundancy and inconsistent detection in large and semi-structered case bases. 1998.

[64] S. O. Rezende, A. G. Evsukoff, A.C.B. Garcia, A. C. P. L. F. Carvalho, A. P. Braga, M. C. Monrad, N. F. F. Ebecken, O. M. Júnior, P. E. M. Almeida, e T. B. Ludemir. *Sistemas Inteligentes - Fundamentos e Aplicações.* Campus, 2003.

[65] S. J. Russel e P. N. Russel. *Inteligência Artificial - Tradução da Segunda Edição.* Campus, 2004.

[66] R. C. Schank. *Conceptual Dependency: A Theory of Natural Language Understanding.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1972.

[67] R. C. Schank. *Dynamic Memory: A Theory of Reminding and Learning in Coputers and People.* Cambridge University Press, New York, 1982.

[68] R. C. Schank. *Dynamic Memory*. Lawrence Erlbaum Associates, New Jersy, 1989.

[69] R. C. Schank e R. Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[70] B. Selman, H. Levesque, e D. Mitchell. Hard and easy distributions of sat problems. *International Conference on Artificial Intelligence*, páginas 459–465, July de 1992.

[71] A. D. Shapiro. *Structured Induction in Expert Systems*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[72] L. R. Simpson. A computer model of case-based reasoning in problem solving: An investigation in the domain of dispute mediation. technical report git-ics-85/18, 1985.

[73] S. Slade. Case-based reasoning. *AI Magazine Spring*, páginas 42–55, 1991.

[74] V. S. Subrahmanian. Towards a theory of evidential reasoning in logic programming. *Logic Colloquim '87*, Spain, July de 1987. The European Summer Meeting of the Association for Symbolic Logic.

[75] K. Sycara. Using case-based reasoning for plan adaptation and repair. *Workshop on CBR*, páginas 425–434, Clearwater Beach, Florida, 1988. DARPA, Morgan Kaufmann Publisher.

[76] G. L. Torres, C. I. A. Costa, e H. C. Martins. *Decision Making System Based on Fuzzy and Paraconsistent Logics*. IOS Press, New Jersey, 2001.

[77] P. Tsaparas. Nearest neighbor search in multidimensional spaces. Relatório Técnico 319-02, Dept. of Computer Science, University of Toronto, 1999.

[78] B. C. Ávila. Representação do conhecimento utilizando frames. Dissertação de Mestrado, Instituto de Ciências Matemáticas de São Carlos - USP, São Carlos, SP, 1991.

[79] B. C. Ávila. *Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança*. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, SP, 1996.

[80] C. G. von Wangenheim e A. von Wangenheim. *Raciocínio Baseado em Casos*. Manole, 2003.

[81] I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann, 1997.

[82] W. H. Wolberg e O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. páginas 9193–9196, USA, 1990. National Academy of Sciences.

[83] J. Zhang. Selecting typical instances in instance-based learning. páginas 470–479, Aberdeen, GB, 1992. Ninth International Machine Learning Conference.

Apêndice

Apêndice A

Características das Bases Utilizadas

Esse apêndice contém a descrição das características próprias que cada uma das nove bases de casos, utilizadas para os testes possuem e apresenta o uso anterior das mesmas. Estas bases foram disponibilizadas pelo UCI [11] (Repository of Machine Learning Databases), da Universidade da Califórnia.

A.1 Características da Base Têmpera

A Base Têmpera foi originalmente doada ao UCI por David Sterling e Wray Buntine. Esta base trata de caractetrísticas da têmpera em metais.

1. Número de Instâncias: 798
2. Número de Atributos: 38 mais o atributo de classe nominal
    - 6 de valores contínuos
    - 3 de valores inteiros (tratados como nominais ou simbólicos no sistema desenvolvido)
    - 29 nominais (ou simbólicos)
3. Informações sobre os Atributos:

Atributo	Valor	Atributo	Valor
family	–,GB,GK,GS,TN,ZA, ZF,ZH,ZM,ZS	phos	P,-
product-type	C, H, G	cbond	Y,-
steel	-,R,A,U,K,M,S,W,V	marvi	Y,-
carbon	contínuo	exptl	Y,-
hardness	contínuo	ferro	Y,-
temper_rolling	-,T	corr	Y,-
condition	-,S,A,X	blue/bright/ varn/clean	B,R,V,C,-
formability	-,1,2,3,4,5	lustre	Y,-
strength	contínuo	jurofm	Y,-
non-ageing	-,N	s	Y,-
surface-finish	P,M,-	p	Y,-
surface-quality	-,D,E,F,G	shape	COIL, SHEET
enamelability	-,1,2,3,4,5	thick	contínuo
bc	Y,-	width	contínuo
bf	Y,-	len	contínuo
bt	Y,-	oil	-,Y,N
bw/me	B,M,-	bore	0000,0500,0600,0760
bl	Y,-	packing	-,1,2,3
m	Y,-	classes	1,2,3,4,5,U
chrom	C,-		

Observe que existe um valor "–"em vários atributos nominais. Esse valor serve para indicar que determinado atributo não é aplicável em determinado caso, portanto não significa um atributo faltante.

4. Valores de atributos faltantes: denotados com "?". Na base, a distribuição do número de instâncias com valores faltantes para cada atributo é a seguinte:

Atributo	Valor Faltante	Atributo	Valor Faltante
1	0	21	791
2	0	22	730
3	70	23	798
4	0	24	796
5	0	25	772
6	675	26	798
7	271	27	793
8	283	28	753
9	0	29	798
10	703	30	798
11	790	31	798
12	217	32	0
13	785	33	0
14	797	34	0
15	680	35	0
16	736	36	740
17	609	37	0
18	662	38	789
19	798	39	0
20	775		

5. Distribuição das Classes:

Nome da Classe	Número de Instâncias
1	8
2	88
3	608
4	0
5	60
U	34

A.2 Características Base Câncer de Mama

A Base Câncer de Mama é outra base disponibilizado no UCI. Obtida da Universidade dos Hospitais de Wisconsin, Madison do Dr. William H. Wolberg e doada por Olvi Mangasarian. Esta base busca identificar a presença de tumores de mama que podem ser: benignos ou malignos.

1. Número de Instâncias: 699
2. Número de Atributos: 10 mais o atributo de classe nominal
    - 9 valores inteiros (tratados como nominais ou simbólicos no sistema desenvolvido)
    - 1 nominal
3. Informações sobre os Atributos:
    - Classe do atributo tem sido movida para a última coluna

Atributo	Domínio
Sample code number	id number
Clump Thickness	1 - 10
Uniformity of Cell Size	1 - 10
Uniformity of Cell Shape	1 - 10
Marginal Adhesion	1 - 10
Single Epithelial Cell Size	1 - 10
Bare Nuclei	1 - 10
Bland Chromatin	1 - 10
Normal Nucleoli	1 - 10
Mitoses	1 - 10
Class	(2 para benign, 4 para malignant)

4. Valores de Atributos Faltantes: 16

   Existem 16 instâncias em grupos de 1 a 6 que contém um único perdido, isto é, indisponível, agora denotados com "?".

5. Distribuição das Classes:

Nome da Classe	Número de Intâncias	Percentual
Benign	458	65.5%
Malignant	241	34.5%

A.2.1 Uso Anterior

A base do Câncer de Mama já foi bastante utilizada no passado em diversos trabalhos por Wolberg *et al*[82] e Zhang [83].

Em Wolberg, a base utilizada possui as seguintes características:

1. Os atributos 2 até 10 têm sido usados para representar instâncias
2. Cada instância pode assumir uma de duas possíveis classes:
   - benigno ou
   - maligno
3. Tamanho do conjunto de dados: somente 369 instâncias
4. Resultados de classificação coletados:
   - uma tentativa somente
5. Dois pares de hiper-planos paralelos descobriu-se ser consistentes com 50% dos dados.
   - correção dos 50% remanescentes do conjunto de dados: 93.5%
6. Três pares de hiper-planos paralelos descobriu-se ser consistentes com 67% dos dados
   - correção dos 33% remanescentes do conjunto de dados: 95,9%

Em Zhang, a base utilizada possui as seguintes características:

1. Tamanho do conjunto de dados
   - somente 369 instâncias

2. Aplicado 4 algoritmos de aprendizagem baseado em instância

    - Resultados de classificação coletados:
        - um vizinho mais próximo: 93.7%
        - treinado em 200 instâncias, testado nas outras 169

3. Interessantes observações

    - usando instâncias só típicas: 92.2% (armazenando só 23.1 instâncias)
    - treinado em 200 instâncias, testado nas outras 169

A.3 Características da Base Dermatologia

A base Dermatologia foi doada ao UCI por Guvenir H. Altay. Esta base faz diagnóstico diferencial de doenças eritemato-escamosas utilizando intervalos com votação.

1. Número de Instâncias: 366

2. Número de Atributos: 34 mais o atributo de classe nominal

    - 33 valores nominais
    - 1 valor linear ou contínuo

3. Informações sobre os Atributos:

4. Atributos Clínicos (estime 0, 1, 2, 3, a menos que caso contrário indicado):

Atributo	Valor	Atributo	Valor
1	erythema	7	follicular papules
2	scaling	8	oral mucosal involvement
3	definite borders	9	knee and elbow involvement
4	itching	10	scalp involvement
5	koebner phenomenon	11	family history, (0 or 1)
6	polygonal papules	34	Age (linear)

5. Atributos Histórico Patológico (estime 0, 1, 2, 3):

Atributo	Valor
12	melanin incontinence
13	eosinophils in the infiltrate
14	PNL infiltrate
15	fibrosis of the papillary dermis
16	exocytosis
17	acanthosis
18	hyperkeratosis
19	parakeratosis
20	clubbing of the rete ridges
21	elongation of the rete ridges
22	thinning of the suprapapillary epidermis
23	spongiform pustule
24	munro microabcess
25	focal hypergranulosis
26	disappearance of the granular layer
27	vacuolisation and damage of basal layer
28	spongiosis
29	san-tooth appearance of retes
30	follicular horn plug
31	perifollicular parakeratosis
32	inflamatory monoluclear infiltrate
33	band-like infiltrate

6. Valores de Atributos Faltantes: 8

   Existem 8 valores faltando no atributo contínuo (idade), indicados com "?"(tratados como indefinido no sistema desenvolvido)

7. Distribuição das Classes

Número da Classe	Classe	Número de Intâncias
1	psoriasis	112
2	seboreic dermatitis	61
3	lichen planus	72
4	pityriasis rosea	49
5	cronic dermatitis	52
6	pityriasis rubra pilaris	20

A.3.1 Uso Anterior

A base Dermatologia foi anteriormente utilizada no trabalho de Demiroz *et al*[20].

A.4 Características da Base Xadrez

A base Xadrez é outra base disponível no UCI. Originalmente fornecida por Alen D. Shapiro e doado por Rob C. Holte. Esta base refere-se a finalização de uma partida de xadrez mostrando a configuração da Torre-Rei x Peão-do-Rei para alcançar esta finalidade; armazenando os valores para verificar a chance das brancas vencerem ou não.

1. Número de Instâncias: 3196
2. Número de Atributos: 36 mais o atributo de classe nominal
    - 36 valores nominais ou simbólicos
3. Resumo Atributos:
    - Classes (2)
        - branco-pode-ganhar (ganhou)
        - branco-não-pode-ganhar (não ganhou, isto é, perdeu)
4. Valores de Atributos Faltantes: nenhum
5. Distribuição das Classes:

Posição	Percentual	Perde ou Ganha
1669	52%	branco pode ganhar
1527	48%	branco não pode ganhar

A.4.1 Uso Anterior

A base Xadrez foi anteriormente utilizada em vários trabalhos por Shapiro [71], Muggleton [55] e Holte *et al*[33].

A.5 Características da Base Reconhecimento de Vinho

A base Reconhecimento de Vinho foi uma base de casos doada ao UCI por Stefan Aeberhard. Esta base faz o reconhecimento de três tipos de vinho da mesma região da Itália, baseado na análise química de treze constituintes do vinho.

1. Número de Instâncias: 178
2. Número de Atributos: 13 mais o atributo de classe nominal
3. Para cada atributo:
    - 13 de valores contínuos
4. Informação sobre os atributos: Nome do Atributo
    - Classe do atributo tem sido movida para a última coluna

    (1) Alcohol

    (2) Malic acid

    (3) Ash

    (4) Alcalinity of ash

    (5) Magnesium

    (6) Total phenols

(7) Flavanoids

(8) Nonflavanoid phenols

(9) Proanthocyanins

(10) Color intensity

(11) Hue

(12) OD280/OD315 of diluted wines

(13) Proline

5. Valores de atributos faltantes: nenhum.

6. Distribuição das Classes: apresenta-se abaixo

Classe	Número	Número de Instâncias por Classe
class	1	59
class	2	71
class	3	48

A.5.1 Uso Anterior

A base Reconhecimento de Vinho foi anteriormente utilizada em vários trabalhos por Stefan Aeberhard e os resultados obtidos são descritos abaixo.

Em Aeberhard *et al*[3] os dados foram usados com muitos outros para comparar vários classificadores. As classes são separáveis, entretanto só o **algoritmo classificador RDA** alcançou 100% de classificação correta como pode ser visto:

Algoritmo	Percentual
RDA	100%
QDA	99.4%
LDA	98.9%
1NN	96.1%

Todos os resultados utilizaram a técnica de *leave-one-out* (deixar um fora).

Em um contexto de classificação, isto é um problema que pode ser visto como estrutura de classes *well behaved* (bem comportadas). Um bom conjunto de dados para testar um novo classificador mas, não muito desafiador.

Em Aeberhard *et al*[2] os dados foram usados para ilustrar a apresentação superior utilizando uma nova função de avaliação com RDA.

A.6 Características da Base Jogo-da-Velha

A base Jogo-da-Velha foi uma base de casos doada ao UCI por David W. Aha.

1. Número de Instâncias: 958
    - cada atributo corresponde a um quadro do jogo
2. Número de Atributos: 9 mais o atributo de classe nominal
    - Todos os atributos podem ter de 1 de 3 valores possíveis
        - x = jogador "x"tomou
        - o = jogador "o"tomou
        - b = branco
        - todos os atributos são nominais
3. Informação sobre os Atributos

Atributo	Valor
top-left-square	x,o,b
top-middle-square	x,o,b
top-right-square	x,o,b
middle-left-square	x,o,b
middle-middle-square	x,o,b
middle-right-square	x,o,b
bottom-left-square	x,o,b
bottom-middle-square	x,o,b
bottom-right-square	x,o,b
Class	positive, negative

4. Valores de atributos faltantes: nenhum.

5. Distribuição das Classes: cerca de 65.3% são positivas, isto é, ganhos para "x".

A.6.1 Uso Anterior

A base Jogo-da-Velha, foi anteriormente utilizada em vários trabalhos por Matheus *et al*[49], [48] e Aha [4].

Matheus *et al*[49], o CITRE (*Constructive Induction on Decision Trees*) foi aplicado utilizando 100 instâncias para o treinamento e 200 instâncias fixas para os testes. Em um estudo que utiliza várias quantias de conhecimento específico de domínio, sua exatidão média mais alta foi 76.7% (utilizando-se da árvore final de decisão resultante dos testes).

Matheus [48] as experiências semelhantes utilizando-se do CITRE inclui aprendizado em curvas utilizando-se de até 500 instâncias fixas para treinamento e utilizando todas as outras instâcias da base para os testes. O alcance de exatidão atingiu 90% mas, valores específicos não são fornecidos.

Aha [4], utilizou as instâncias da seguinte forma: 70% para treinamento e 30% para os testes. Avaliou mais de dez testes e obteve os seguintes resultados divulgados para seis algoritmos:

Algoritmo	Resultado
NewID	84.0%
CN2	98.1%
MBRtalk	88.4%
IB1	98.1%
IB3	82.0%
IB3-CI	99.1%

Os resultados também mostram que quando adiciona-se uns 10 atributos irrelevantes de valor-ternário; relativamente espera-se semelhantes resultados, exceto no algoritmo IB1's cujo desempenho degrada mais rapidamente que nos outros.

A.7 Características da Base Íris

A base Íris foi uma base de casos doada ao UCI por From Fisher.

1. Número de Instâncias: 150 (50 em cada uma das três classes)
2. Número de Atributos: 4 mais o atributo de classe nominal
    - 4 numéricos
3. Informação sobre os atributos:

Atributo	Valor
sepal length	in cm
sepal width	in cm
petal length	in cm
petal width	in cm
class	Iris Setosa, Iris Versicolour, or Iris Virginica

4. Valores de atributos faltantes: nenhum.
5. Sumário de Estatísticas:

Type	Min	Max	Mean	SD	Class Correlation
sepal length:	4.3	7.9	5.84	0.83	0.7826
sepal width:	2.0	4.4	3.05	0.43	-0.4194
petal length:	1.0	6.9	3.76	1.76	0.9490 (high)
petal width:	0.1	2.5	1.20	0.76	0.9565 (high)

6. Distribuição das Classes: 33.3% para cada uma das três classe

A.7.1 Uso Anterior

A base Íris, foi anteriormente utilizada em vários trabalhos por Fisher [24], por Dasarathy [19], por Gates [26], [27] e por Duda *et al*[22].

Dasarathy obteve o seguinte resultado: taxas de classificações extremamente baixas (0% para classe setosa).

Gates também obteve resultados com taxas de classificações muito baixas.

A.8 Características da Base Zoológico

A base Zoológico foi uma base de casos doada ao UCI por Richard S. Forsyth.

1. Número de Instâncias: 101
2. Número de Atributos: 17 mais o atributo de classe nominal
    - 15 booleanos (tratados como nominais ou simbólico no sistema desenvolvido)
    - 2 numéricos
3. Informação sobre os atributos:

Atributo	Valor
animal name	Unique for each instance
hair	Boolean
feathers	Boolean
eggs	Boolean
milk	Boolean
airborne	Boolean
aquatic	Boolean
predator	Boolean
toothed	Boolean
backbone	Boolean
breathes	Boolean
venomous	Boolean
fins	Boolean
legs	Numeric (set of values: 0,2,4,5,6,8)
tail	Boolean
domestic	Boolean
catsize	Boolean
type	Numeric (integer values in range [1,7])

4. Valores de atributos faltantes: nenhum.

5. Distribuição das Classes:

Nome da Classe	Conjunto de Animais
41	aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby, wolf
20	chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren
5	pitviper, seasnake, slowworm, tortoise, tuatara
13	bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
4	frog, newt, toad
8	flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
10	clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

A.8.1 Uso Anterior

A base Zoológico, não possui nenhum uso anterior diferente do que é mostrado no guia do usuário Forsyth's PC-BEAGLE.

A.9 Características da Base Sobrevivência de Haberman

A base Sobrevivência de Haberman foi doada ao UCI por Tjen-Sien Lim. Este conjunto de dados contém casos de um estudo que foi realizado entre 1958 e 1970 no Hospital Billing's da Universidade de Chicago nos pacientes sobreviventes após realização de cirurgias para câncer de mama.

1. Número de Instâncias: 306

2. Número de Atributos: 3 mais o atributo de classe nominal

    - 3 numéricos

3. Informação sobre os atributos:

Atributo	Valor
Age of patient at time of operation	numerical
Patient's year of operation	year minus 1900 numerical
Number of positive auxiliary nodes detected	numerical
Survival status class atribute	1 = the patient survived 5 years or longer 2 = the patient died within 5 year

4. Valores de atributos faltantes: nenhum.

A.9.1 Uso Anterior

A base Sobrevivência de Haberman foi anteriormente utilizada em diversos trabalhos por Haberman [29], por Landwehr *et al*[43] e por Lo [45].