

Estratégias de adaptação de casos em  
sistemas de raciocínio baseado em casos

*Cláudio Adriano Policastro*

*orientador: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho*

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

**“VERSÃO REVISADA APÓS A DEFESA”**

USP – São Carlos  
Fevereiro/2004

Aos meus pais, Hugo e Amélia,  
Aos meus irmãos, César e Márcio,  
À minha esposa, Patrícia.

Esta Dissertação foi preparada com o formatador de textos  $\text{\LaTeX}$ . O sistema de citações de referências bibliográficas utiliza o padrão *Chicago* do sistema  $\text{BIB}\text{\TeX}$ .

© Copyright 2004 por Claudio Adriano Policastro  
Todos os Direitos Reservados

# Agradecimentos

---

Ao Prof. André Carlos Ponce de Leon Ferreira de Carvalho, pela confiança, incentivo, apoio e paciência na orientação deste trabalho e principalmente pela amizade.

Ao Prof. Alexandre Cláudio Botazzo Delbem, pela amizade, incentivo e apoio.

Aos Profs. Maria Carolina Monard, Alneu de Andrade Lopes e Eduardo do Valle Simões, pela amizade e incentivo.

Aos amigos do Labic e de outros laboratorios, pelo companheirismo, amizade e apoio.

Aos professores e funcionários do ICMC, pela amizade.

À toda minha família e em especial aos meus pais e minha esposa, pelo incentivo, apoio e dedicação.



## Resumo

---

Raciocínio Baseado em Casos é uma metodologia para a resolução de problemas baseado em experiências passadas. Essa metodologia tenta solucionar um novo problema recuperando e adaptando soluções previamente conhecidas de problemas similares. Porém, cada solução recuperada, em geral, requer adaptações para que possa ser utilizada como solução de um novo problema. Portanto, a adaptação de casos é uma característica desejável em sistemas de Raciocínio Baseado em Casos. Um dos maiores desafios da área de RBC é o desenvolvimento de métodos eficientes para a adaptação de casos. Em contraste com a aquisição de casos, o conhecimento para adaptação não é facilmente disponibilizado e é de difícil obtenção (Hanney, 1996; Wiratunga et al., 2002).

A forma de adaptação mais utilizada é a codificação de regras de adaptação, demandando um significativo esforço para a aquisição de conhecimento (Hanney, 1996).

Uma alternativa para superar as dificuldades associadas à aquisição de conhecimento para adaptação de casos tem sido a utilização de abordagens híbridas e de algoritmos de aprendizado automático para a aquisição do conhecimento utilizado para a adaptação.

Este trabalho investiga a utilização de abordagens híbridas para adaptação de casos empregando algoritmos de Aprendizado de Máquina. As abordagens aprendem o conhecimento necessário para a adaptação de casos automaticamente a partir de uma base de casos e aplicam esse conhecimento para realizar a adaptação de soluções recuperadas.

# Abstract

---

Case Based Reasoning is a methodology for problem solving based on past experiences. This methodology tries to solve a new problem by retrieving and adapting previously known solutions of similar problems. However, retrieved solutions, in general, require adaptations in order to be applied to new contexts. One of the major challenges in Case Based Reasoning is the development of an efficient methodology for case adaptation. In contrast to case acquisition, knowledge for case adaptation is not easily available and is hard to obtain (Hanney, 1996; Wiratunga et al., 2002).

The most widely used form of adaptation employs handcoded adaptation rules, which demands a significant effort of knowledge acquisition for case adaptation, presenting a few difficulties (Hanney, 1996).

An alternative to overcome the difficulties associated to the acquisition of knowledge for case adaptation has been the use of hybrid approaches and automatic learning algorithms for the acquisition of the knowledge used for the adaptation.

This work investigates the use of hybrid approaches for case adaptation employing Machine Learning algorithms. The approaches automatically learn adaptation knowledge from a case base and apply it to adapt retrieved solutions.

# Sumário

---

Sumário . . . . .	iii
Lista de Figuras . . . . .	v
Lista de Tabelas . . . . .	viii
Lista de Algoritmos . . . . .	ix
Lista de Abreviaturas e Siglas . . . . .	xi
<b>1 Introdução</b>	<b>1</b>
<b>2 Raciocínio Baseado em Casos</b>	<b>5</b>
2.1 Considerações Iniciais . . . . .	5
2.2 Motivação . . . . .	6
2.3 História do Raciocínio Baseado em Casos . . . . .	8
2.4 Repositórios de Conhecimento . . . . .	9
2.4.1 Representação de Casos . . . . .	10
2.4.2 Indexação de Casos . . . . .	12
2.5 O Ciclo de Raciocínio Baseado em Casos . . . . .	12
2.5.1 Recuperação de Casos . . . . .	14
2.5.2 Reutilização de Casos . . . . .	16
2.5.3 Revisão de Casos . . . . .	16
2.5.4 Retenção de Casos . . . . .	17
2.6 Manutenção de Bases de Casos . . . . .	18
2.7 Considerações Finais . . . . .	19
<b>3 Adaptação de Casos</b>	<b>21</b>
3.1 Considerações Iniciais . . . . .	21
3.2 Adaptação por Substituição . . . . .	22
3.3 Adaptação por Transformação . . . . .	28
3.4 Adaptação por Geração . . . . .	30
3.5 Considerações Finais . . . . .	31

<b>4</b>	<b>Sistemas Híbridos de RBC</b>	<b>33</b>
4.1	Considerações Iniciais . . . . .	33
4.2	Integração Neurosimbólica . . . . .	34
4.2.1	Abordagem Unificada . . . . .	34
4.2.2	Abordagem Híbrida . . . . .	35
4.3	Integração Simbólico-Evolutiva . . . . .	37
4.4	RBC Híbrido . . . . .	38
4.4.1	Abordagem Híbrida . . . . .	38
4.4.2	Abordagem Unificada . . . . .	43
4.5	Considerações Finais . . . . .	44
<b>5</b>	<b>Algoritmos e Métodos Utilizados</b>	<b>45</b>
5.1	Considerações Iniciais . . . . .	45
5.2	Aprendizado de Máquina . . . . .	45
5.2.1	Aprendizado Indutivo . . . . .	46
5.3	Pré-processamento de Dados . . . . .	46
5.3.1	Tratamento de Valores Desconhecidos . . . . .	47
5.3.2	Tratamento de Classes Desbalanceadas . . . . .	47
5.4	Algoritmos . . . . .	48
5.4.1	Rede ART2 . . . . .	48
5.4.2	Algoritmo C4.5 . . . . .	49
5.4.3	Algoritmo M5 . . . . .	50
5.4.4	Rede MLP . . . . .	52
5.4.5	Algoritmo SVM . . . . .	53
5.4.6	Algoritmo K-Vizinhos mais Próximos . . . . .	55
5.4.7	Algoritmo Kd-Tree . . . . .	56
5.5	Análise de Resultados . . . . .	57
5.6	Considerações Finais . . . . .	59
<b>6</b>	<b>Abordagem Híbrida de Adaptação</b>	<b>61</b>
6.1	Considerações Iniciais . . . . .	61
6.2	Abordagem Híbrida de Adaptação de Casos . . . . .	62
6.2.1	Geração dos Padrões de Adaptação . . . . .	63
6.2.2	Mecanismo de Adaptação de Casos . . . . .	64
6.3	Comitês de Algoritmos de AM . . . . .	65
6.4	Abordagem Híbrida de Recuperação de Casos . . . . .	67
6.4.1	Processo de Indexação . . . . .	68
6.4.2	Processo de Recuperação . . . . .	68
6.5	Descrição dos Conjuntos de Dados . . . . .	69
6.6	Descrição dos Experimentos . . . . .	72
6.7	Resultados Empíricos . . . . .	76

<i>Sumário</i>	iii
6.8 Análise dos Resultados . . . . .	83
6.9 Considerações Finais . . . . .	84
<b>7 Conclusão</b>	<b>99</b>
7.1 Considerações Iniciais . . . . .	99
7.2 Discussão . . . . .	99
7.3 Trabalhos Futuros . . . . .	101
7.4 Publicações . . . . .	102
<b>Referências Bibliográficas</b>	<b>111</b>

## Lista de Figuras

---

---

2.1	Representação de casos . . . . .	11
2.2	Ciclo de Raciocínio Baseado em Casos . . . . .	13
3.1	Classificação das estratégias de adaptação de casos. . . . .	21
3.2	Uma árvore semântica de papéis funcionais de objetos. . . . .	23
4.1	Estratégias de integração neurosimbólica . . . . .	34
4.2	Estratégias de integração simbólico-evolutiva. . . . .	37
4.3	Estratégias de integração de RBC . . . . .	38
5.1	Arquitetura da rede ART2 . . . . .	50
5.2	Exemplo de arquitetura de uma árvore de decisão. . . . .	51
5.3	Exemplo de arquitetura do <i>Model Tree</i> . . . . .	52
5.4	Arquitetura da rede MLP . . . . .	53
5.5	Arquitetura de um classificador SVM . . . . .	55
5.6	Particionamento do espaço dos dados e estrutura da Kd-Tree resultante . . . . .	57
6.1	Exemplo do processo de geração dos padrões de adaptação. . . . .	65
6.2	Arquitetura dos comitês investigados. Os comitês (a), (b) e (c) foram utilizados para a adaptação de componentes discretos. Os comitês (d), (e) e (f) foram utilizados para a adaptação de componentes contínuos. . . . .	67
6.3	Arquitetura do Mecanismo de Indexação e Recuperação de Casos. . . . .	68
6.4	Distribuição dos valores das soluções dos conjuntos de dados com valores de saída contínuos. . . . .	73
6.5	Metodologia dos experimentos realizados. . . . .	77

## Lista de tabelas

---

5.1	Características Gerais dos Sistemas de Aprendizado. . . . .	46
6.1	Estrutura dos Casos de Servo Mecanismo. . . . .	70
6.2	Estrutura dos Casos de Carros Importados. . . . .	71
6.3	Estrutura dos Casos de Moradia. . . . .	72
6.4	Estrutura dos Casos de Pontes de Pittsburgh. . . . .	72
6.5	Estrutura dos Casos de Qualidade de Água. . . . .	73
6.6	Estrutura dos Casos da Base de Vinho. . . . .	73
6.7	Características dos Conjuntos de Dados. . . . .	74
6.8	Topologias das Redes Neurais Artificiais Empregadas como Com- binadores. . . . .	74
6.9	Topologias das Redes Neurais Artificiais Empregadas como Esti- madores. . . . .	75
6.10	Resultados obtidos. . . . .	83
6.11	Resultados do conjunto de dados Servo. . . . .	86
6.12	Valores do Domínio para o Conjunto de Dados Servo. . . . .	87
6.13	Resultados do Teste $t$ para o conjunto de dados Servo. . . . .	87
6.14	Resultados do conjunto de dados Imports. . . . .	88
6.15	Valores do Domínio para o Conjunto de Dados Imports. . . . .	89
6.16	Resultados do Teste $t$ para o conjunto de dados Imports. . . . .	89
6.17	Resultados do conjunto de dados Housing. . . . .	90
6.18	Valores do Domínio para o Conjunto de Dados Housing. . . . .	91
6.19	Resultados do Teste $t$ para o conjunto de dados Housing. . . . .	91
6.20	Resultados do conjunto de dados Bridges. . . . .	92
6.21	Resultados do Teste $t$ para o conjunto de dados Bridges. . . . .	93
6.22	Resultados do conjunto de dados Water. . . . .	94
6.23	Valores do Domínio para o Conjunto de Dados Water. . . . .	95
6.24	Resultados do Teste $t$ para o conjunto de dados Water. . . . .	95
6.25	Resultados do conjunto de dados Vinho. . . . .	96

6.26 Valores do Domínio para o Conjunto de Dados Vinho. . . . . 97  
6.27 Resultados do Teste  $t$  para o conjunto de dados Vinho. . . . . 97



# Lista de Algoritmos

---

---

1	Versão básica do algoritmo <i>K-Vizinhos mais Próximos</i> para problemas com função objetivo de valores discretos . . . . .	56
2	Geração de Padrões de Adaptação. . . . .	63
3	Mecanismo de Adaptação de Casos. . . . .	64
4	Indexação de Novos Casos. . . . .	68
5	Recuperação de Casos. . . . .	69

## Lista de Abreviaturas e Siglas

---

---

- AG** Algoritmo Genético
- AM** Aprendizado de Máquina
- ART2** Teoria de Ressonância Adaptativa
- BC** Base de Casos
- IA** Inteligência Artificial
- MLP** *Perceptron* Multi - Camadas
- RBC** Raciocínio Baseado em Casos
- RBF** Função de Base Radial
- RNA** Rede Neural Artificial
- SVM** Máquina de Vetores de Suporte

---

## Introdução

---

Raciocínio Baseado em Casos (RBC) é uma metodologia para a resolução de problemas baseada na utilização de experiências passadas. RBC procura resolver um novo problema utilizando um processo de recuperação e adaptação de soluções de outros problemas similares previamente solucionados. O estudo do paradigma de RBC é dirigido por duas motivações principais (Leake, 1996):

- a primeira, da Ciência Cognitiva, é o desejo de modelar o comportamento humano;
- a segunda, da Inteligência Artificial (IA), é o desejo de desenvolver uma técnica que torne os sistemas de IA mais efetivos.

O interesse em RBC como um modelo cognitivo é apoiado por estudos do raciocínio humano que evidenciam o raciocínio a partir de casos em vários contextos. As observações de que as pessoas raciocinam baseado em casos também impulsionaram o interesse em RBC como uma técnica da IA (Leake, 1996). Seres humanos são robustos solucionadores de problemas. Eles rotineiramente solucionam problemas difíceis aplicando conhecimento muitas vezes impreciso ou limitado, melhorando seu desempenho com a experiência. Todas essas qualidades são desejáveis em sistemas de IA aplicados ao mundo real.

Quando sistemas de RBC são aplicados para a resolução de problemas reais, as soluções recuperadas raramente podem ser utilizadas diretamente como solução de um novo problema. Elas precisam ser adaptadas para serem aplicadas a novas situações. Portanto, a adaptação de casos é uma característica desejável em sistemas de RBC. A adaptação de casos pode ser tão simples

como uma substituição de um componente de uma solução, ou tão complexa como a mudança completa da estrutura da solução.

A adaptação de casos é um dos maiores desafios em RBC e é uma área de pesquisa em aberto na qual ainda existe muito a ser feito. Em contraste com conteúdo dos casos, o conhecimento para a adaptação do casos não é facilmente disponibilizado e é difícil de ser adquirido (Hanney, 1996; Wiratunga et al., 2002). A maioria das abordagens propostas para adaptação de casos são específicas para o domínio alvo e requerem um considerável esforço para aquisição do conhecimento necessário para a sua utilização.

Assim, é necessário o desenvolvimento de abordagens de adaptação de casos que facilitem a aquisição do conhecimento necessário para o seu funcionamento, bem como de abordagens de adaptação mais genéricas possam ser aplicadas em classes de domínios de problemas e utilizem um conjunto reduzido do conhecimento do domínio.

Uma alternativa para superar essas dificuldades associadas à aquisição do conhecimento necessário para adaptação de casos tem sido a utilização de abordagens híbridas e de aquisição automática de conhecimento por um mecanismo de aprendizado (Hanney e Keane, 1997; Puvis e Athalye, 1997; Gómez et al., 1999; Garza e Maher, 2000; Corchado et al., 2001; Lotfy e Sameh, 2002; Jarmulak et al., 2001; Craw et al., 2001; Wiratunga et al., 2002).

A utilização de outras técnicas de IA para compor o ciclo de raciocínio de RBC pode fornecer sistemas inteligentes mais eficientes e com melhor desempenho em diversas aplicações.

Este trabalho investiga a utilização de abordagens híbridas de adaptação de casos que utilizam algoritmos de Aprendizado de Máquina (AM) como mecanismo de adaptação. Este trabalho investiga, também, a utilização de algoritmos para a aquisição automática de conhecimento de adaptação de casos. Com isso, busca-se explorar características que permitam a construção de abordagens de adaptação que possam ser aplicadas em diferentes domínios e que reduzam o esforço necessário para a aquisição do conhecimento de adaptação de casos.

Neste trabalho, são descritas as abordagens de adaptação de casos investigadas e os algoritmos construídos para a implementação dessas. São apresentados diversos experimentos realizados para a avaliação empírica das abordagens investigadas, utilizando seis conjuntos de dados. Os experimentos realizados avaliam indiretamente o conhecimento automaticamente adquirido e utilizado pelas abordagens de adaptação de casos, comparando o desempenho dessas abordagens com classificadores empregados individualmente sobre o mesmo problema.

---

Este trabalho está organizado da seguinte forma. No Capítulo 2, são introduzidos os principais conceitos do paradigma de RBC. No Capítulo 3, são discutidas as estratégias e abordagens de adaptação de casos encontradas na literatura. No Capítulo 4, são descritas as estratégias de integração de diferentes paradigmas de IA para a construção de sistemas híbridos, em particular, sistemas híbridos de RBC. No Capítulo 5 são discutidos brevemente os algoritmos e métodos utilizados para a realização dos experimentos contidos neste trabalho. No Capítulo 6 são apresentadas as abordagens investigadas e os resultados obtidos nos experimentos realizados. No Capítulo 7 são feitas as considerações finais deste trabalho.



---

## Raciocínio Baseado em Casos

---

### 2.1 *Considerações Iniciais*

RBC é uma metodologia para a resolução de problemas utilizando experiências passadas. RBC procura resolver um novo problema apresentado ao sistema recuperando problemas previamente solucionados de uma memória ou Base de Casos (BC) e adaptando a solução associada a cada problema recuperado para atender aos requisitos do novo problema.

A literatura mostra que RBC pretende contribuir não apenas com uma tecnologia específica, mas com uma metodologia consistente de raciocínio e aprendizado (Kolodner, 1992; Aamodt e Plaza, 1994; Watson, 1999; Watson, 2002).

RBC difere de outros paradigmas da IA em muitos aspectos (Aamodt e Plaza, 1994):

- em contraste com paradigmas que somente utilizam conhecimento geral do domínio ou constroem relações entre problemas e soluções, RBC é capaz de utilizar conhecimento específico de problemas prévios;
- RBC não apenas denota uma metodologia de raciocínio, ele denota um paradigma de aprendizado de máquina que possibilita o aprendizado incremental, pela atualização da BC sempre que um novo problema é resolvido, tornando o novo conhecimento disponível para utilizações futuras.

O paradigma de RBC é baseado em dois princípios sobre a natureza do mundo (Leake, 1996). O primeiro princípio dita que o mundo é regular: problemas similares possuem soluções similares. Conseqüentemente, as soluções de problemas prévios similares são um bom ponto de partida para a

solução de novos problemas. O segundo princípio dita que os tipos de problemas encontrados por um agente tendem a recorrer. Assim, novos problemas tendem a ser similares a problemas prévios.

De acordo com a forma de utilização do conhecimento capturado nos casos, os sistemas de RBC podem ser divididos em duas classes (Kolodner, 1992; Leake, 1996):

- *RBC interpretativo*: utiliza os problemas prévios como ponto de referência para classificação ou caracterização de novas situações. No RBC interpretativo, o objetivo é formar um julgamento sobre, ou classificação de, um novo problema, pela comparação e contraste com casos que já foram julgados ou classificados;
- *RBC solucionador de problemas*: utiliza os problemas prévios para sugerir soluções que possam ser aplicadas a novas circunstâncias. O objetivo do RBC solucionador de problemas é utilizar uma solução prévia para gerar a solução para o novo problema.

Neste capítulo é introduzido o paradigma de Raciocínio Baseado em Casos e os elementos componentes dessa metodologia de raciocínio. Na Seção 2.2 são apresentadas as principais motivações da área pesquisa de RBC. Na Seção 2.3 são apontados alguns marcos da história do paradigma de RBC. Na Seção 2.4 é apresentada uma discussão sobre o conteúdo dos repositórios de conhecimento de um sistema de RBC. Na Seção 2.5 é discutido o ciclo de raciocínio do paradigma de RBC. Na Seção 2.6 é abordado o problema da manutenção da BC em sistemas reais. Na Seção 2.7 são apresentadas as considerações finais deste capítulo.

## 2.2 Motivação

O estudo do paradigma de RBC é dirigido por duas motivações principais (Leake, 1996):

- a primeira, da Ciência Cognitiva, é o desejo de modelar o comportamento humano;
- a segunda, da IA, é o desejo de desenvolver uma técnica que torne os sistemas de IA mais efetivos na solução de problemas reais.

O interesse em RBC como um modelo cognitivo é suportado por estudos do raciocínio humano que evidenciam o raciocínio a partir de casos em vários contextos. As observações de que as pessoas raciocinam baseado em casos



também impulsionaram o interesse em RBC como uma técnica da IA (Leake, 1996).

Seres humanos são robustos solucionadores de problemas. Eles rotineiramente solucionam problemas difíceis aplicando conhecimento muitas vezes impreciso ou limitado e seu desempenho melhora com a experiência. Todas essas qualidades são desejáveis em sistemas de IA aplicados ao mundo real.

Diversos autores (Kolodner, 1992; Leake, 1996; Watson e Marir, 1994; Main et al., 2001) têm discutido que, quando utilizado em situações apropriadas, o paradigma de RBC pode amenizar diversos problemas da IA, tais como:

- aquisição de conhecimento;
- manutenção de conhecimento;
- eficiência na resolução de problemas;
- qualidade das soluções;

A seguir, esses problemas são discutidos em maiores detalhes.

### *Aquisição de Conhecimento*

O paradigma de RBC trabalha sobre problemas e soluções específicos, tornando desnecessária a decomposição e a generalização de experiências em forma de regras. Alguns domínios são favoráveis para aplicação do paradigma de RBC, por apresentarem registros de problemas previamente solucionados, facilitando assim a aquisição de conhecimento. Nesses domínios, o custo da aquisição de conhecimento para um sistema de RBC pode ser baixo.

Porém, nem todos os domínios são favoráveis à aplicação do paradigma de RBC. Existem domínios em que casos prévios podem não estar disponíveis ou estar disponíveis em uma forma complexa para seu uso. Nessas situações, a aplicação de RBC pode depender de um significativo esforço de aquisição de conhecimento e não apresentar vantagens sobre outros paradigmas.

Existem ainda domínios nos quais a aquisição de regras é mais fácil que a aquisição de casos. Para esses domínios, um sistema tradicional baseado em regras provavelmente trabalhará melhor que um sistema de RBC.

### *Manutenção do Conhecimento*

A definição da base de conhecimento inicial é apenas o primeiro passo no desenvolvimento de sistemas baseados em conhecimento. O entendimento inicial do problema muitas vezes é imperfeito e incompleto, requerendo que o conhecimento inicial seja refinado. O paradigma de RBC oferece um significativo benefício para a manutenção de conhecimento adquirido. O usuário do

sistema pode ser habilitado a adicionar à BC aqueles casos que julgar necessário, sem a intervenção do engenheiro de conhecimento (Leake, 1996).

### *Eficiência na Resolução de Problemas*

Em geral, os seres humanos alcançam um desempenho satisfatório no processo de resolução de problemas, apesar de os problemas comumente encontrados diariamente serem NP-completos<sup>1</sup>. O reuso de soluções prévias ajuda a aumentar a eficiência do processo de resolução de problemas, uma vez que o raciocínio parte de uma solução prévia parcialmente correta, ao invés de repetir novamente todo o esforço do raciocínio.

### *Qualidade das Soluções*

Quando os princípios do domínio não estão bem entendidos, as regras podem ser imperfeitas. Nessas situações, as soluções sugeridas por casos podem ser mais precisas que aquelas sugeridas por encadeamento de regras. Casos capturam conhecimento específico sobre a solução de um problema em uma dada circunstância.

## *2.3 História do Raciocínio Baseado em Casos*

As raízes do RBC em IA vêm dos trabalhos de Roger Schank (1982) em memória dinâmica e o papel central que a lembrança de situações prévias (casos ou episódios) e os padrões de situações têm na resolução de problemas e aprendizado (Aamodt e Plaza, 1994).

O primeiro sistema que pode ser chamado de raciocinador baseado em casos é o CYRUS (1983), desenvolvido por Janet Kolodner, na Universidade de Yale (grupo de pesquisa de Roger Schank).

Outro importante grupo de pesquisa em RBC foi o de Bruce Porter (1986), na Universidade do Texas, Austin. Eles inicialmente focaram o problema de aprendizado de máquina em tarefas de classificação. Isto levou ao desenvolvimento do sistema PROTOS, que deu ênfase a integração do conhecimento geral do domínio com o conhecimento específico dos casos.

Na Europa, as pesquisas em RBC foram iniciadas pouco depois que nos Estados Unidos. Os trabalhos em RBC eram fortemente ligados ao desenvolvimento de sistemas especialistas e as pesquisas sobre aquisição de conhecimento. Um dos primeiros trabalhos europeus sobre RBC foi o sistema

---

<sup>1</sup>problemas para os quais não existem, até o momento, algoritmos capazes de fornecer uma solução precisa em tempo polinomial.

MOLKE, construído por Michael Richter e Klaus Dieter, juntamente com outros pesquisadores, na Universidade de Kaiserslautern.

Na Universidade de Trondheim (1989), Agnar Aamodt, juntamente com outros pesquisadores, estudou os aspectos gerais de aprendizado de RBC e da manutenção do conhecimento. Também foi focado o uso combinado do conhecimento geral do domínio e do conhecimento específico dos casos prévios.

Em 1994, as pesquisas sobre RBC nos Estados Unidos, bem como na Europa, tiveram um rápido crescimento, que pode ser verificado pelo grande número de conferências e artigos publicados em congressos e nas revistas de IA.

De acordo com (Main et al., 2001; Malek, 2001; Jarmulak et al., 2001; Wiratunga et al., 2002; Plaza e Arcos, 2002; Prentzas e Hatzilygeroudis, 2002; Falkman, 2002), as atividades em RBC estão focadas em problemas como:

- adaptação de casos;
- utilização de métricas de adaptabilidade durante a recuperação dos casos;
- sistemas híbridos que combinam RBC com outras técnicas de IA;
- manutenção de eficiência do sistema durante seu ciclo de vida;
- manutenção de conhecimento.

## 2.4 Repositórios de Conhecimento

O conhecimento necessário para os processos do paradigma de RBC foi identificado em termos de quatro repositórios de conhecimento (Richter, 1995). Esses repositórios são:

- o vocabulário que descreve o domínio do problema;
- a base de casos;
- o conhecimento para a determinação da similaridade entre casos;
- o conhecimento necessário para a adaptação de soluções.

Não existe separação intrínseca entre esses quatro repositórios, com respeito ao processo geral de resolução de problemas. Isto é, cada repositório pode manter qualquer parte do conhecimento necessário para a resolução de um novo problema.

A seguir, algumas considerações sobre o conhecimento necessários aos processos de raciocínio em RBC são apresentadas em maiores detalhes.

### 2.4.1 Representação de Casos

Um sistema de RBC é dependente da estrutura e conteúdo de sua coleção de casos, também denominada estrutura de memória. O problema de representação é primariamente o problema de decidir o que armazenar em um caso, encontrar uma estrutura apropriada para descrever o conteúdo dos casos e decidir como a memória de casos deve ser organizada e indexada para possibilitar a recuperação e a reutilização de soluções prévias. Um problema adicional é a decisão de como deve ser a integração da memória de casos com o conhecimento geral do domínio (Aamodt e Plaza, 1994). Alguns autores (Cunningham e Bonzano, 1999; Varma, 1999) apontam a análise e definição da estrutura e representação dos casos como a tarefa mais complexa do trabalho de aquisição de conhecimento para um sistema de RBC.

RBC possibilita uma grande flexibilidade de representação do conhecimento. Dependendo dos atributos dos problemas a serem representados, uma estrutura conveniente pode ser selecionada. O objetivo de um sistema de RBC influencia fortemente o conteúdo de seus casos (Main et al., 2001). Um caso pode representar muitos tipos de conhecimento e pode possuir diferentes formas de representação. Um caso deve conter uma certa riqueza de informações e uma certa complexidade em sua estrutura de representação (Aamodt e Plaza, 1994). Porém, quando não existem especialistas disponíveis no domínio da aplicação, ou quando não existe um modelo que possa ser extraído do domínio do problema, os casos são usualmente representados como um registro de pares atributo - valor, divididos em dois conjuntos: problema e solução.

Em algumas situações, casos podem ser divididos em sub-casos e organizados em uma representação hierárquica ou uma representação orientada a objeto.

Os casos devem armazenar atributos que garantam que o caso será recuperado no contexto apropriado. Durante a escolha do formato de representação dos casos, alguns aspectos devem ser considerados:

- se os casos possuem segmentos que formam naturalmente sub-casos ou componentes e se a estrutura interna de um caso pode ser acomodada na forma de representação selecionada para os casos;
- se o conteúdo ou os atributos que formam o caso possuem tipos ou estruturas associadas;
- se a linguagem selecionada para implementar o sistema irá limitar os formatos que podem ser utilizados na representação do caso;
- o mecanismo de indexação e busca que será utilizado;

- a forma como os casos estão disponíveis ou podem ser obtidos no domínio da aplicação.

O formato de representação dos casos e da BC deve facilitar a recuperação de casos prévios em momentos apropriados. Várias abordagens de organização para BC têm sido utilizadas (Sporn et al., 1999; Main et al., 2001). Memória Plana e Estrutura Hierárquica são duas das abordagens mais comuns (Malek e Amy, 1994). A escolha do modelo de memória para representação dos casos deve considerar alguns fatores:

- a representação utilizada para a base de casos;
- o propósito para o qual o sistema está sendo desenvolvido;
- o número e complexidade dos casos que serão armazenados;
- o número de atributos que serão utilizados para recuperação dos casos prévios;
- se algumas características são similares o suficiente para serem agrupadas;
- o conhecimento que se tem sobre o domínio da aplicação para determinar as similaridades entre os casos.

Alguns exemplos de representação e casos são mostrados na Figura 2.1.

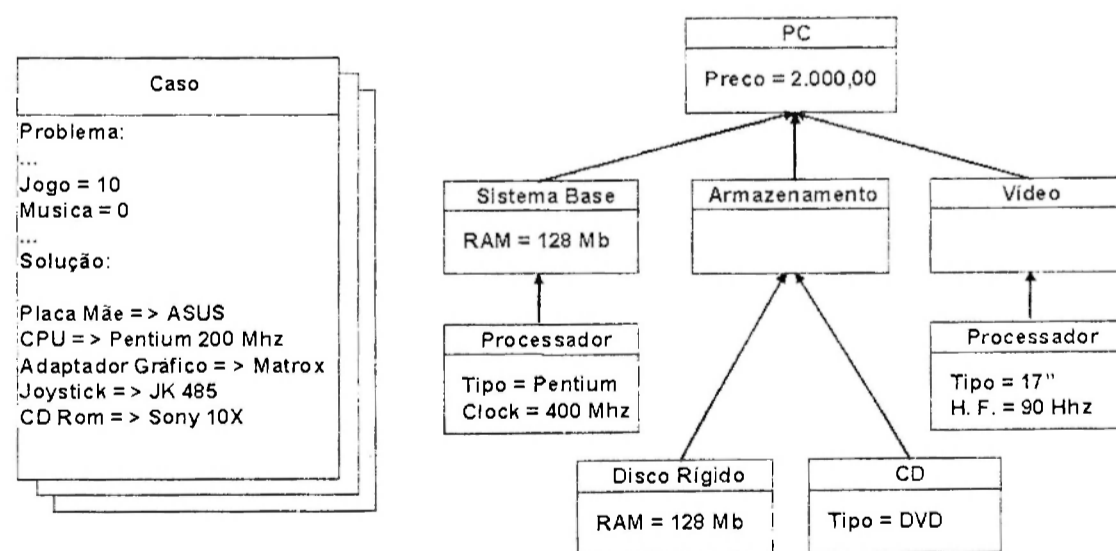


Figura 2.1: Representação de casos: (a) como dois conjuntos de pares atributo - valor e (b) em uma estrutura orientada a objeto (Bergmann e Stahl, 1998).

### 2.4.2 Indexação de Casos

Indexação refere-se à atribuição de índices aos casos prévios para futuras comparações e recuperações (Aamodt e Plaza, 1994).

Os índices dos casos formam um conjunto de atributos — da descrição do problema — que são relevantes para a predição da utilidade do caso em uma dada situação. A suposição é que quanto maior a similaridade entre descritores de uma consulta e os atributos que formam os índices dos casos de uma BC, mais úteis esses casos serão para a solução do problema atual (Wangenheim, 2000).

A construção de índices apropriados torna possível a recuperação dos casos adequados para um problema com o qual o sistema é confrontado. Isto porque os índices de um caso irão determinar o contexto no qual esse caso deve ser recuperado em situações futuras. Portanto, os índices dos casos precisam ser úteis e predizíveis. Além disso, índices precisam ser suficientemente abstratos para possibilitar que os casos apropriados possam ser recuperados em várias situações futuras. Porém, índices não podem ser muito abstratos, para que os casos não sejam recuperados em situações impróprias, exigindo um maior esforço para adaptação da solução prévia recuperada.

Técnicas indutivas de indexação e de extração de características permitem que os engenheiros de conhecimento encontrem atributos relevantes para a indexação dos casos e os organizem em uma estrutura eficiente de memória.

## 2.5 O Ciclo de Raciocínio Baseado em Casos

Um modelo largamente aceito do processo de Raciocínio Baseado em Casos é o Ciclo de RBC formalizado por Aamodt e Plaza (Aamodt e Plaza, 1994), mostrado na Figura 2.2, o qual compreende quatro tarefas principais:

1. recuperar: O termo *recuperar* foi adotado em (Aamodt e Plaza, 1994) para denominar a fase de recuperação do caso armazenado na BC mais similar ao novo problema apresentado.
2. reutilizar: O termo *reutilizar* foi adotado em (Aamodt e Plaza, 1994) para denominar a fase de adaptação da solução do caso recuperado, que consiste na reutilização de uma cópia da solução prévia como ponto de partida para a construção da solução do novo problema. No restante desse trabalho é adotado o termo adaptação de casos.
3. revisar: O termo *revisar* foi adotado em (Aamodt e Plaza, 1994) para denominar a fase na qual a solução proposta pela fase de reutilização ou adaptação.

- reter: O termo *revisar* foi adotado em (Aamodt e Plaza, 1994) para denominar a fase na qual o novo problema solucionado pode ser armazenado na BC.

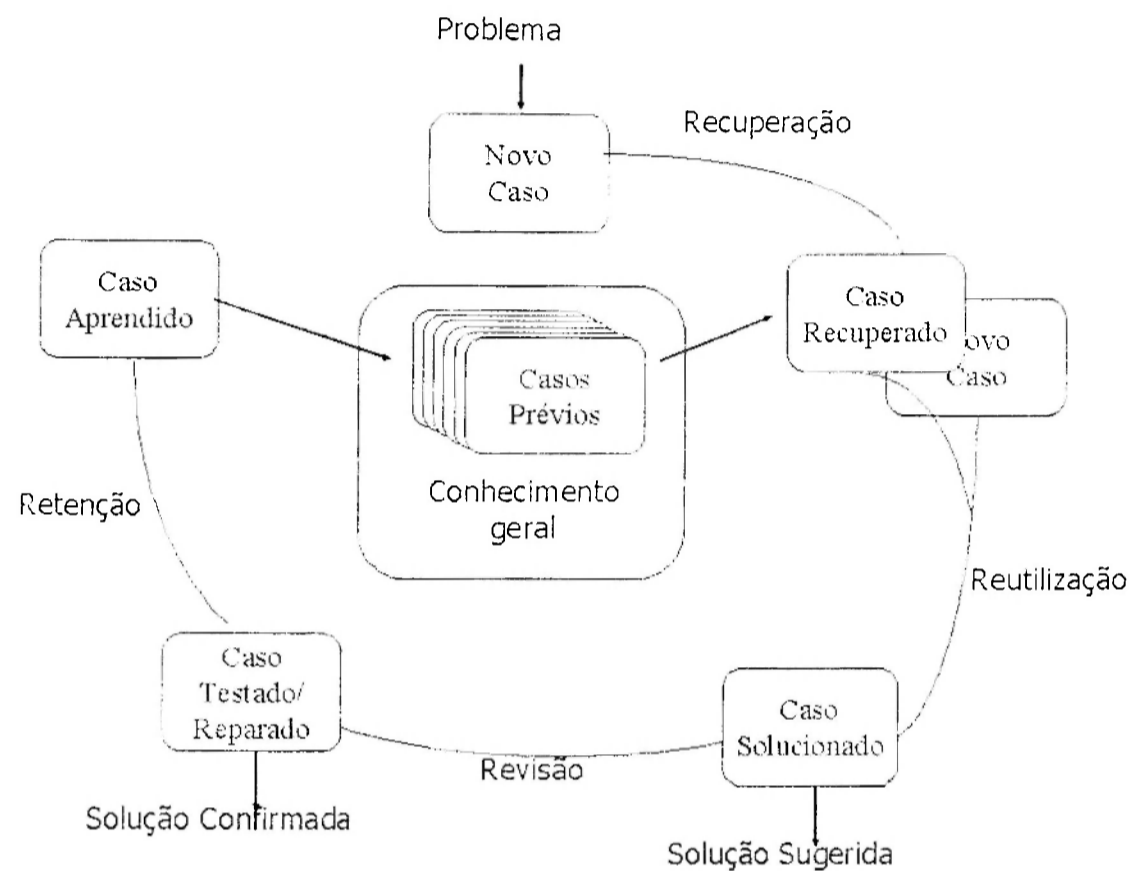


Figura 2.2: Ciclo de Raciocínio Baseado em Casos (Aamodt e Plaza, 1994).

A descrição de um problema define um novo caso. Esse novo caso é utilizado para a recuperação de um caso da coleção de casos prévios (BC).

O caso recuperado é combinado com o novo caso durante o processo de reutilização, originando um caso resolvido, isto é, uma solução proposta para o problema inicial.

Durante o processo de revisão, essa solução é testada para validação: por aplicação no mundo real; por meio de uma simulação; pela avaliação do usuário ou utilizando o conhecimento da própria BC.

Após o processo de revisão, essa solução é reparada em caso de necessidade.

Durante a fase de retenção, a informação útil é armazenada para uso futuro e a BC é alterada pela inclusão de conhecimento em forma de um novo caso ou pela alteração de um caso existente (Aamodt e Plaza, 1994).

No centro do Ciclo de RBC está o conjunto de casos relatando as situações e experiências passadas e, possivelmente, conhecimento geral do domínio. O

conhecimento do domínio pode ser utilizado em várias fases do processo de RBC, como, por exemplo, na análise da similaridade entre o novo problema e os casos da BC, na adaptação da solução recuperada e na validação da solução proposta para um novo problema.

### 2.5.1 Recuperação de Casos

O objetivo da fase de recuperação é recuperar um caso prévio da BC que seja útil à resolução de um problema atual. Entretanto, é muito difícil determinar se um caso é útil para um problema específico ou medir se determinado caso é mais útil do que outro para a resolução de um dado problema. Um caso pode ser considerado útil ao problema atual se ele pode ser facilmente aplicado para a resolução desse novo problema. Entretanto, isto só pode ser determinado quando se tenta utilizar a solução do caso recuperado para solucionar o problema atual. Esse problema é resolvido em RBC assumindo que situações ou problemas similares requerem soluções similares (Leake, 1996). Nos últimos anos foram propostas algumas abordagens de recuperação de casos que utilizam métricas de adaptabilidade para recuperar os casos que tornem o processo de adaptação mais eficiente (Smyth e Keane, 1998).

O conceito de medida de similaridade é a chave para a realização da busca por casos na BC. Uma medida de similaridade associa um valor numérico a um caso recuperado, expressando seu grau de similaridade com um novo caso. Essas medidas são baseadas em uma interpretação geométrica, na qual os casos são considerados pontos em um espaço  $n$ -dimensional. Cada dimensão corresponde a um atributo do conjunto de índices dos casos, que são um sub-conjunto dos atributos da descrição do problema. O conceito mais simples de similaridade entre um novo caso e um caso recuperado é a medida da distância euclidiana entre os pontos no espaço  $n$ -dimensional correspondentes ao novo caso e ao caso recuperado, respectivamente (Wangenheim, 2000).

A tarefa central do processo de recuperação é, portanto, recuperar da BC o caso mais similar ao problema atual, dada uma BC e uma consulta descrevendo o novo problema, de acordo com uma medida de similaridade entre os atributos que formam o conjunto de índices do novo problema e o caso recuperado (Wangenheim, 2000).

O primeiro passo no processo de recuperação é a representação do problema corrente, que é normalmente feita por meio da atribuição dos valores dos atributos que formam o conjunto dos índices dos casos. Adicionalmente, o processo de representação do problema pode envolver um pré-processamento dos valores atribuídos, utilizando conhecimento geral do domínio ou um modelo do domínio da aplicação (Wangenheim, 2000). A seleção dos casos similares é então feita utilizando os valores atribuídos aos atributos que formam os



índices dos casos.

Nos últimos anos, várias técnicas foram propostas para serem utilizadas na implementação do processo de recuperação de casos (Main et al., 2001; Spori et al., 1999), incluindo:

- *kd-trees*: kd-tree é uma generalização multi-dimensional das árvores de busca binária de uma dimensão. Essa técnica permite a integração de RBC com Algoritmos de Aprendizado Indutivos e possibilitam uma recuperação por similaridade baseada em Árvores de Decisão (Bentley, 1975).
- *Fish and Shrink*: Fish and Shrink possibilita a redução sucessiva dos intervalos de similaridades possíveis entre uma consulta e os casos de uma BC por meio de alguns testes explícitos, possibilitando uma rápida busca em domínios nos quais os casos possuam uma estrutura de representação complexa (Schaaf, 1996).
- *Case Retrieval Nets*: Case Retrieval Nets utilizam conceitos de redes neurais para a construção de uma estrutura de memória contendo nós armazenando valores dos atributos que formam os índices dos casos. Os nós são conectados por ligações com pesos que indicam a similaridade entre os valores armazenados nos nós. Essa técnica utiliza um processo de ativação dos nós para selecionar os casos a serem recuperados (Lenz, 1996b).
- *Footprint-Based Retrieval*: Footprint-Based Retrieval é um mecanismo de recuperação de casos que utiliza durante o processo de recuperação um conjunto de casos denominado *Footprint*. Esse conjunto de casos é um conjunto mínimo de casos que representa todos os casos de uma BC. Esse mecanismo de recuperação atua em dois passos. Primeiro, o mecanismo de recuperação compara o novo problema apresentado com o problema contido em cada caso pertencente ao conjunto *Footprint*, investigando assim a região mais promissora do espaço das soluções da BC. Em seguida, o mecanismo de recuperação compara o novo problema apresentado com os casos pertencentes ao agrupamento de casos da BC que está associado ao caso mais similar encontrado no conjunto *Footprint* (Smyth e McKenna, 1999).
- *Redes Neurais Artificiais*: podem ser utilizadas em domínios de aplicação com representações de casos simples ou complexas e com grandes BCs. Para tal, as redes procuram por padrões de similaridade entre os casos e agrupam os casos armazenados em grupos similares.

Baseado na similaridade determinada para cada caso, o melhor caso é recuperado. Após a recuperação do caso, o sistema de RBC pode solicitar informação adicional ou uma confirmação do usuário do sistema.

### 2.5.2 Reutilização de Casos

Uma vez recuperado o caso mais similar ao problema atual, o sistema de RBC tenta reutilizar a solução do caso recuperado para construir a solução para o novo problema. Se necessário, a solução recuperada é adaptada pelo sistema para atender aos requerimentos do novo problema.

A maneira mais simples de reutilizar uma solução recuperada é simplesmente copiar essa solução e utilizá-la como solução do novo problema. Entretanto, quando forem recuperados casos semelhantes ao problema atual, dificilmente se conseguirá aplicar diretamente a solução recuperada a esse novo problema, devido às diferenças existentes entre o caso recuperado e o atual. Sendo assim, a solução terá que ser ajustada através de um processo de adaptação que corrija essas diferenças.

O processo de adaptação inclui a identificação das diferenças entre o problema prévio recuperado e o problema atual, e a adaptação da solução recuperada para uma nova solução que satisfaça a situação atual. A adaptação em situações simples pode ser realizada, por exemplo, por ajuste de parâmetros, regras e fórmulas, ou métodos de configuração de acordo com regras e restrições (Wangenheim, 2000).

Muitos sistemas de RBC efetuam adaptação de casos para domínios de problemas específicos, codificando o conhecimento necessário para o processo de adaptação na forma de um conjunto de regras de adaptação. As regras de adaptação são então aplicadas ao caso recuperado para que esse possa atender a nova situação. Alguns sistemas de RBC utilizam pedaços de casos existentes na memória para executar a adaptação. A extensão do processo de adaptação de um sistema de RBC depende do seu domínio de aplicação.

Na maioria dos sistemas de RBC propostos, as adaptações de soluções recuperadas são feitas manualmente pelo usuário do sistema. Em muitas aplicações, o custo da captura do conhecimento necessário para a adaptação automática é alto (Leake, 1996; Hanney, 1996; Wangenheim, 2000; Wiratunga et al., 2002).

### 2.5.3 Revisão de Casos

O objetivo da fase de revisão é avaliar a solução criada na fase de reutilização. A revisão da solução pode ser feita:

- pelo usuário do sistema;

- por meio de uma simulação;
- pela aplicação da solução ao ambiente real.

A revisão da solução pode também ser feita utilizando-se o conhecimento da própria BC (Aamodt e Plaza, 1994; Leake, 1996; Wangenhein, 2000).

Se a revisão executada for bem sucedida, o novo caso pode ser retido na BC. Quando uma solução gerada na fase de reutilização não está correta, essa solução é reparada pelo sistema, utilizando-se o conhecimento geral do domínio, ou pelo usuário (Aamodt e Plaza, 1994). O reparo de casos envolve um processo de detecção de falhas da solução proposta e a recuperação ou a geração de explicações para elas. As explicações sobre as falhas podem ser armazenadas e utilizadas para evitar que essas ocorram novamente.

#### 2.5.4 Retenção de Casos

A retenção é o processo de incorporação do novo caso na BC para futuras utilizações. Essa fase completa o Ciclo de RBC (Wangenhein, 2000).

A fase de retenção tem dois propósitos principais (Main et al., 2001):

1. quanto mais informação é adicionada à BC, uma maior similaridade entre os casos recuperados e um novo problema pode ser alcançada;
2. adicionar informação à BC melhora a solução que o sistema pode fornecer.

A retenção pode ocorrer de muitas maneiras. A adição de um novo caso com a solução para um problema é o método mais comum. Um outro método de retenção utilizado em sistemas de RBC consistem em avaliar a solução gerada para modificar os índices dos casos armazenados ou para modificar os critérios de recuperação de casos.

Segundo (Aamodt e Plaza, 1994), o processo de retenção envolve:

- a seleção de informação gerada durante o processo de raciocínio e construção do caso a ser retido;
- a montagem do índice desse novo caso para futuras recuperações;
- a integração desse novo caso na BC existentes.

Além de casos solucionados, as falhas, isto é, as informações do processo de revisão, podem também ser extraídas e retidas separadamente como casos falhos. Quando um novo problema é apresentado ao sistema, as passas prévias associadas ao processo de solução de problemas similares podem ser utilizadas para evitar que essas ocorram novamente (Aamodt e Plaza, 1994).

Alguns sistemas de RBC recorrem ao usuário do sistema ou a políticas de manutenção de conhecimento para decidir se o novo caso deve ou não ser retido. Essas políticas são discutidas a seguir.

## 2.6 Manutenção de Bases de Casos

A manutenção em RBC pode ser definida como o estabelecimento de políticas para a revisão da organização e do conteúdo (representação, cobertura do domínio e informações quantitativas) da BC, de maneira a facilitar futuros raciocínios, para um conjunto particular de objetivos de desempenho e eficiência (Leake e Wilson, 1998). Ela pode envolver a revisão da BC em nível de implementação (Leake e Wilson, 1998) (modificando atributos, estrutura dos casos e estrutura organizacional da base) ou a revisão em nível do conhecimento representado (Smyth, 1998) (casos desatualizados, inconsistentes ou redundantes precisam ser excluídos; grupos de casos podem ser mesclados para eliminar redundância e aumentar o poder de raciocínio do sistema; casos podem ser re - editados para reparar inconsistências).

Quando sistemas de RBC são aplicados à resolução de problemas, novos casos adicionados à BC podem tornar-se importantes e casos prévios podem tornar-se redundantes ou conflitantes com os novos casos adicionados. O crescimento da BC pela adição de novos casos pode também reduzir o desempenho e a eficiência do sistema.

Para superar esses problemas, foram propostas nos últimos anos várias políticas para adição e exclusão de casos da BC (Smyth e Keane, 1995; Smyth, 1998; Zhu e Yang, 1999; Leake e Wilson, 2000), e políticas para revisão de casos e BCs (Racine e Yang, 1997; Reinartz et al., 2000). Em (Reinartz et al., 2000) é proposto um novo Ciclo de RBC composto por seis tarefas principais: recuperar, reutilizar, revisar, reter, rever e restaurar. As três primeiras fases formam o ciclo de raciocínio. As três últimas formam a fase de manutenção do conhecimento do sistema.

Leake e Wilson (Leake e Wilson, 1998), estabeleceram um *framework* para categorizar as abordagens de manutenção em termos das políticas de manutenção, que determinam quando e como um sistema de RBC executa a manutenção de sua BC. As políticas de manutenção são descritas, nesse *framework*, em termos de como elas coletam informações relevantes para o processo de manutenção, como elas decidem quando disparar o processo de manutenção, os tipos de operações de manutenção disponíveis e como as operações de manutenção selecionadas são executadas.

## 2.7 Considerações Finais

Neste Capítulo foram apresentados os principais conceitos do paradigma de Raciocínio Baseado em Casos.

Foi apresentado também um modelo amplamente aceito do processo de Raciocínio Baseado em Casos, formalizado por Aamodt e Plaza (Aamodt e Plaza, 1994), nomeado Ciclo de RBC.

Também foi visto que, quando utilizado em situações apropriadas, o paradigma de RBC pode amenizar alguns problemas encontrados em outros paradigmas da IA. Deve-se ressaltar que o paradigma de RBC não é a resposta para todos os problemas encontrados IA. Existem situações em que RBC não pode ser aplicado para a resolução de problemas, situações em que RBC não constitui a solução mais adequada e situações em que RBC deve ser aplicado em conjunto com outras técnicas.



## Adaptação de Casos

### 3.1 Considerações Iniciais

Quando sistemas de RBC são aplicados para a resolução de problemas, as soluções recuperadas raramente podem ser utilizadas diretamente como solução de um novo problema. As soluções recuperadas precisam ser adaptadas para serem aplicadas às novas situações. Portanto, a adaptação de casos é uma importante característica em sistemas de RBC. A adaptação de casos pode ser tão simples como uma substituição de um componente de uma solução, ou tão complexa como a mudança da estrutura da solução.

A adaptação pode ocorrer de várias formas: um novo item pode ser incluído na solução, um item da solução pode ser removido, algum item pode ser substituído por outro, ou alguma parte da solução prévia pode ser transformada.

Várias estratégias de adaptação de casos foram propostas na literatura (Kolodner, 1993; Smyth e Cunningham, 1993). Uma possível classificação dessas estratégias é ilustrada na Figura 3.1.

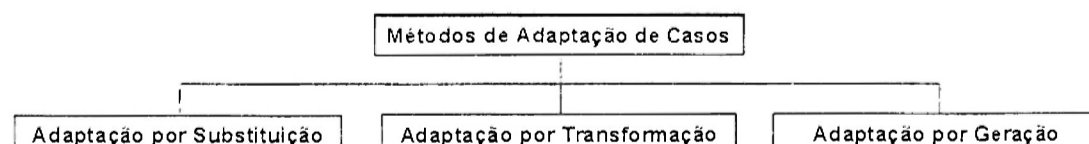


Figura 3.1: Classificação das estratégias de adaptação de casos.

As estratégias de adaptação por substituição trocam os valores de uma solução prévia por valores apropriados para uma nova situação.

As estratégias de adaptação por transformação alteram a estrutura da solução pela inclusão ou remoção de componentes da solução recuperada para

atender às necessidades do novo problema.

As estratégias de adaptação por geração repetem os passos que foram executados para se obter uma solução recuperada, no contexto do novo problema. Quando um novo problema é apresentado, a solução desse novo problema é obtida diretamente repetindo-se o processo executado para a obtenção da solução do problema recuperado.

Neste capítulo são introduzidas as estratégias e abordagens para adaptação de casos mais empregadas em sistemas de RBC. Na Seção 3.2 são descritas as estratégias de adaptação por substituição. Na Seção 3.3 são apresentadas as estratégias de adaptação por transformação. Na Seção 3.4 são discutidas as estratégias de adaptação por geração. Na Seção 3.5 são feitas as considerações finais desse capítulo.

### 3.2 Adaptação por Substituição

Substituição é o processo de escolha e troca de componentes de uma solução recuperada por novos componentes. A substituição pode ocorrer de diferentes formas:

- um componente de uma solução prévia pode ser substituído;
- uma parte de um componente de uma solução pode ser substituída;
- um grupo completo de componentes ou partes pode ser substituído.

A adaptação por substituição é uma das formas mais utilizadas para a adaptação de casos. Várias estratégias podem ser utilizadas para esse tipo de adaptação. A seguir, são comentadas as estratégias de adaptação por substituição encontradas na literatura.

#### *Reinstanciação*

Reinstanciação é uma estratégia de adaptação de casos que pode ser aplicada quando a estrutura da solução do novo problema e do caso recuperado permanece igual e os componentes da solução do novo problema possuem os mesmos papéis funcionais dos componentes da solução do caso recuperado. Para a aplicação dessa estratégia de adaptação, o sistema de RBC precisa implementar mecanismos para identificar a correspondência entre os papéis funcionais<sup>1</sup> dos vários componentes e que permitam obter a estrutura genérica da solução do caso recuperado (Kolodner, 1993).

<sup>1</sup>O papel funcional de um componente em uma solução de um problema consiste na funcionalidade ou ação que esse componente desempenha na solução.



A reinstanciação torna-se simples quando a estrutura das soluções dos problemas do domínio é conhecida e quando os papéis funcionais são especificados na representação do problema. Nos casos mais simples, a reinstanciação é um processo de substituição de valores prévios por novos, instanciando uma estrutura prévia com novos valores.

Porém, quando um domínio de aplicação não possibilita que o sistema de RBC mantenha a mesma estrutura de solução entre um novo caso e um caso recuperado, ou quando um sistema de RBC não é capaz de relacionar os papéis funcionais dos componentes da nova solução e da solução recuperada, a adaptação de casos por reinstanciação pode ser dificultada ou até impossibilitada (Kolodner, 1993).

A correspondência de papéis funcionais entre componentes de um caso recuperado e de um novo problema pode ser computada utilizando uma estrutura de conhecimento e abstração (árvore semântica, por exemplo), como a mostrada na Figura 3.2. Nessa Figura, os objetos filhos de um mesmo pai possuem o mesmo papel no mundo (Kolodner, 1993).

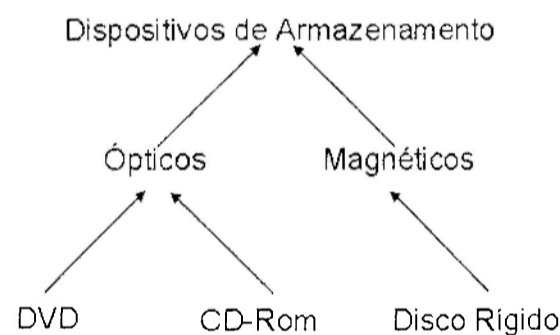


Figura 3.2: Uma árvore semântica de papéis funcionais de objetos.

Quando os métodos para a computação de abstrações e correspondências para o domínio do problema tornam-se complexos, a adaptação por reinstanciação pode não ser eficiente e não é indicada (Kolodner, 1993). Após a adaptação por reinstanciação, passos adicionais de adaptação podem ser necessários para tratar conflitos ou inconsistências que podem resultar da substituição dos componentes da solução recuperada.

Em (Hammond, 1986) é apresentado um planejador baseado em casos que utiliza reinstanciação para adaptar as soluções recuperadas para solucionar um novo plano. Quando um novo problema é apresentado, descrito por um conjunto de metas, o sistema recupera um plano que possui um conjunto de metas semelhantes às do novo problema. Essa recuperação é auxiliada por estruturas de conhecimento que permitem a generalização e a especificação de conceitos do domínio de aplicação do sistema. O sistema utiliza essas estruturas para verificar quais metas do caso recuperado e do novo problema

possuem papéis funcionais ou ações semelhantes dentro da solução do problema e recupera o caso prévio com maior semelhança entre seus conjuntos de metas. Após a recuperação de uma solução, o sistema utiliza as estruturas auxiliares de conhecimento para efetuar a reinstanciação da solução recuperada, substituindo as metas do caso recuperado pelas metas do novo problema. Após a reinstanciação, passos adicionais são executados para tratar as exceções ou particularidades das novas metas introduzidas na solução.

### *Ajuste de Parâmetros*

Ajuste de parâmetros é uma estratégia de adaptação de casos que utiliza interpolação de valores entre o caso recuperado e um novo problema.

A estratégia de ajuste de parâmetros modifica os componentes de uma solução recuperada em resposta às diferenças entre as descrições do problema do caso recuperado e de um novo problema. Primeiro, as descrições do novo problema e do problema contido no caso recuperado são comparadas e suas diferenças são extraídas. Em seguida, heurísticas especiais de ajuste são aplicadas à solução recuperada para criar uma nova solução. As heurísticas de ajuste capturam o relacionamento entre os atributos do problema e os componentes da solução (Kolodner, 1993).

A aplicação de heurísticas de ajuste requer uma comparação detalhada entre as descrições do novo problema e do problema contido no caso recuperado. O sistema de RBC precisa implementar algum controle sobre esse processo de comparação para que esse processo se torne eficiente. Para atender a esse problema, um sistema de RBC precisa implementar mecanismos que permitam: a manutenção de um *checklist* de comparações que sempre podem ser feitas antes do ajuste de parâmetros, ou a manutenção de uma lista de heurísticas de ajustes de parâmetros onde cada heurística é indexada pelas circunstâncias nas quais pode ser aplicada, de acordo com as comparações que ela requer.

A manutenção de um *checklist* é apropriado em domínios em que as comparações entre os casos são sempre feitas sobre as mesmas dimensões, ou atributos do problema, e não se alteram.

A manutenção de uma lista de heurísticas é apropriada para dividir o conjunto de possibilidades quando a execução de todas as comparações torna-se proibitiva.

Em (Wiratunga et al., 2002) é proposto um método indutivo de aquisição de conhecimento para adaptação de casos utilizando como fonte de conhecimento a própria BC do sistema de RBC. O método consiste de um algoritmo que executa comparações entre pares de casos e extrai as diferenças das descrições de seus problemas e soluções. Esse algoritmo utiliza as diferenças

extraídas dos pares de casos, juntamente com a descrição do problema do caso utilizado como problema atual, para a construção de um conjunto de dados de padrões que serão utilizadas para o treinamento de um algoritmo de aprendizado indutivo. O método proposto utiliza o algoritmo boosting (Freund e Schapire, 1996) para treinar um comitê composto pelo algoritmo indutivo Rise (Domingos, 1996) – um algoritmo híbrido que combina indução e aprendizado baseado em instâncias – utilizando o conjunto de dados gerado. Para a aplicação do conhecimento de adaptação, foi construído um sistema de RBC que utiliza o comitê de algoritmos Rise na fase de adaptação de casos. Quando um novo problema é apresentado ao sistema, um caso prévio é recuperado e o sistema de RBC utiliza o comitê de algoritmos para indicar as ações de adaptação sobre o caso recuperado.

A aplicação das estratégias de adaptação por ajuste de parâmetros também pode ser feita por meio de um algoritmo de AM. Para isso, o algoritmo deve ser capaz de aproximar uma função que pode interpolar valores para a solução do novo problema.

Em (Corchado et al., 1998; Corchado et al., 2001), uma Rede Neural Artificial (RNA) do tipo Função de Base Radial (RBF) (Orr, 1996) é utilizada durante a fase de adaptação de casos. Quando um novo problema é apresentado ao sistema, um conjunto de casos prévios similares é recuperado e esses casos são utilizados para treinar uma rede RBF. Após o treinamento da rede com o conjunto de casos recuperados, a descrição do novo problema é apresentada para a rede e essa age como uma função fornecendo a solução para o problema atual.

Deve-se ressaltar que o ajuste de parâmetros é apenas uma estratégia de interpolação. Ela não pode criar uma nova solução, mas somente ajustar valores de uma solução existente em resposta às diferenças entre o caso recuperado e um novo problema.

#### *Busca em Memória*

Existem situações onde é necessário somente substituir pequenas partes de uma solução recuperada para a construção de uma nova solução. Nessas situações, o uso de estratégias de adaptação de casos, como a reinstanciação, que substitui um conjunto de componentes de uma vez, não é indicado. A busca em memória é uma estratégia de adaptação de casos que consiste em realizar uma busca em uma estrutura auxiliar de conhecimento e abstração (rede semântica, por exemplo) por algum componente similar que possa substituir um componente da solução atual (Kolodner, 1993). Essa estratégia é geralmente utilizada quando a solução do caso recuperado está quase correta para um novo problema e pode ser ajustada somente com pequenas

substituições.

Para efetuar a busca por um componente que possa substituir um componente de uma solução recuperada, o sistema de RBC deve construir uma especificação parcial do componente que poderia levar a uma boa substituição e ativar um mecanismo de busca para iniciar uma nova busca na memória.

A busca em memória pode tornar-se mais eficiente por meio da implementação de mecanismos que forneçam, além da descrição do componente que deve ser encontrado, instruções sobre como encontrar o componente necessário para a substituição da solução recuperada. Esses mecanismos podem permitir que o sistema encontre respostas rapidamente iniciando a busca em locais apropriados da memória. Para a aplicação dessa estratégia de adaptação, o sistema de RBC precisa implementar um conjunto de heurísticas de busca especializadas e mecanismos para a definição de critérios de aplicabilidade associados a cada heurística definida (Kolodner, 1993).

As heurísticas de busca devem ser definidas de forma a serem úteis para encontrar todos os tipos de componentes que podem ser regularmente requisitados, e que tenham um método bem conhecido para serem encontrados. Em adição, o sistema de RBC precisa fornecer um conjunto de guias que mostrem quando utilizar cada heurística. A busca em memória é tão eficiente quanto a heurística de especialização utilizada na busca. Portanto, os critérios utilizados para definir quando cada heurística deve ser utilizada devem ser efetivos e robustos.

Um sistema de RBC proposto em (Leake, 1995; Leake et al., 1996) possui, inicialmente, somente um pequeno conjunto de regras de transformação e métodos de busca em memória. Quando é apresentado um novo problema ao sistema, ele recupera o caso prévio mais similar e transfere esse caso recuperado para o mecanismo de adaptação. O mecanismo de adaptação seleciona uma transformação para aplicar sobre o caso recuperado e, em seguida, ativa uma busca em memória para encontrar as informações necessárias para operacionalizar a regra de transformação. O sistema melhora sua capacidade de adaptação aplicando RBC ao próprio processo de adaptação: seqüências de passos de adaptação são retidas para reuso quando surgem novos problemas de adaptação similares aos previamente retidos. Quando o processo autônomo de adaptação falha, o sistema pode ainda permitir que o usuário efetue a adaptação manualmente e reter a seqüência de passos da adaptação executada pelo usuário em um novo caso de adaptação.

Em (Falkman, 2002) é apresentada uma abordagem de adaptação utilizando um modelo declarativo. A adaptação é dada por um processo computacional interativo. O modelo disponibiliza dois tipos de conhecimento para a adaptação de casos: conhecimento sobre as possíveis adaptações e conheci-

mento sobre quando e como uma adaptação específica deve ser empregada. As definições de adaptação  $D_A$  definem que um valor  $v_i$  pode ser adaptado para um valor  $v_m$ . Os métodos de adaptação  $m_A$  definem como e quando realizar uma adaptação. A adaptação é realizada em um processo iterativo que utiliza o conhecimento definido em  $D_A$  e em  $m_A$  para substituir de maneira incremental os componentes da solução recuperada até obter a solução para um novo problema.

#### *Substituição Baseada em Casos*

Em algumas situações, o melhor lugar para se procurar um componente substituto é em outro caso similar armazenado na BC. A substituição baseada em casos é uma estratégia de adaptação de casos que busca por um componente substituto em casos prévios que possam sugerir uma alternativa e então extrai e utiliza essa alternativa para substituir o componente prévio (Kolodner, 1993). Existem dois passos para o processo de substituição baseada em casos:

1. encontrar partes de casos similares que possam sugerir alternativas à parte que precisa ser substituída;
2. escolher um substituto entre as sugestões.

No primeiro passo, casos apropriados são recuperados da BC. No segundo passo, suas alternativas são extraídas, avaliadas, e uma é selecionada. Para a aplicação dessa estratégia de adaptação, o sistema de RBC precisa implementar mecanismos que permitam a construção de uma especificação do componente que precisa ser encontrado e mecanismos que permitam que os casos sejam indexados de maneira a possibilitar a recuperação baseada no conteúdo de suas partes.

Uma possibilidade é dividir os casos em partes e então indexar essas partes separadamente. Outra possibilidade é indexar os casos tanto para as situações que eles solucionam como para as situações em que suas partes devem ser recuperadas.

Em (Hinkley e Toomey, 1994) é apresentado um sistema de RBC que utiliza a adaptação baseada em casos para a construção de novas soluções. O sistema é aplicado ao domínio de autoclave e as soluções propostas pelo sistema são conjuntos layouts de peças para serem curadas. Quando uma solução recuperada possui partes ou peças com layouts inapropriados para o novo problema apresentado, o sistema busca por partes de outros casos armazenados que possam substituir as partes inapropriadas. A suposição é que as partes inapropriadas podem ser substituídas por partes de outras soluções

armazenadas que estejam localizadas em lugares similares e com layouts semelhantes àquelas que estão sendo substituídas. O sistema constrói uma especificação da parte que precisa ser encontrada (incluindo a localização, o tipo de material e o layout da parte) e recupera partes apropriadas das soluções armazenadas na BC para adaptar a solução recuperada.

### 3.3 Adaptação por Transformação

Estratégias de adaptação por substituição não podem ser utilizadas para a inclusões ou remoções de componentes da solução recuperada. Elas são aplicáveis somente se já existe um componente ou conceito que pode ser substituído por algum outro mais apropriado.

Para as situações nas quais a modificação exige a inclusão ou a remoção de componentes da solução, deve-se utilizar um processo que transforme a estrutura da solução, incluindo ou removendo componentes. As estratégias de transformação alteram uma solução recuperada para atender a uma nova situação (Kolodner, 1993):

- pela inclusão ou remoção de componentes;
- pela inclusão ou remoção de partes dos componentes.

Algumas estratégias de transformação utilizam heurísticas de senso comum para alterar valores que necessitam de transformação. Outras estratégias de transformação utilizam raciocínio baseado em um modelo causal para sugerir uma adaptação que pode ser utilizada para transformar a estrutura da solução ou alguns de seus valores (Kolodner, 1993).

A adaptação com heurísticas de senso comum utiliza um pequeno conjunto de heurísticas que empregam conhecimento sobre a importância relativa dos diferentes componentes de um item para determinar se remoções ou substituições devem ser executadas (Kolodner, 1993). Existem alguns requisitos para se efetuar transformações de senso comum:

- o sistema deve ser capaz de identificar as partes dos componentes que necessitam de adaptação;
- as representações dos casos devem separar os componentes primários e secundários, isto é, componentes que definem a solução e componentes que somente dão suporte ou que caracterizam os componentes que definem a solução;
- as representações dos casos devem manter o relacionamento interno entre os componentes da solução, permitindo que se determine como a

remoção ou inclusão de componentes afeta os demais componentes existentes.

A adaptação utilizando modelos baseia-se no conhecimento das conexões de um modelo causal de um dado domínio ou situação. Nessas estratégias, as heurísticas utilizam o conhecimento causal para a transformação da solução prévia. As heurísticas de adaptação baseadas em modelos são heurísticas de propósito geral e são baseadas no conhecimento de causalidade. Modelos causais particulares acessados pelas heurísticas de propósito geral servem para especializar essas heurísticas (Kolodner, 1993).

Na abordagem proposta em (Smyth e Keane, 1998), a adaptação de casos é feita por uma transformação em dois estágios: primeiro, especialistas gerais de adaptação são empregados para transformar os componentes da solução recuperada no componente desejado para a solução do novo problema; em seguida, estratégias gerais de adaptação são utilizadas para manipular as interações problemáticas que surgem das atividades dos especialistas. Os especialistas de adaptação correspondem a pacotes de conhecimento para a transformação dos componentes da solução, em que cada especialista executa uma tarefa específica de adaptação de acordo com o componente que necessita ser transformado. As estratégias de adaptação correspondem a pacotes de conhecimento que reparam problemas de interações que podem surgir após a aplicação de um ou mais pacotes especialistas. As estratégias de adaptação detectam e reparam diferentes classes de problemas de interações que possam surgir.

A aplicação das estratégias de adaptação por transformação também pode ser feita por meio de um Algoritmo Genético (AG). Por exemplo, as transformações podem ser efetuadas pela aplicação do operador genético de cruzamento. Em (Gómez et al., 1999; Puvis e Athalye, 1997; Garza e Maher, 2000), um AG é utilizado para a adaptação de casos em um sistema de RBC. Nessas abordagens, os casos são adaptados de maneira incremental e em paralelo até que uma solução satisfatória para o novo problema seja construída.

Esse método de adaptação suporta dois tipos de adaptação: paramétrica e estrutural. A adaptação paramétrica é alcançada pelo operador genético de mutação. A adaptação estrutural é alcançada por meio do operador genético de cruzamento. Quando um novo problema é apresentado ao sistema, vários casos satisfazendo parcialmente a solução do problema são recuperados. Esses casos recuperados alimentam o AG, ou seja, formam a população inicial de cromossomos. As adaptações das soluções produzidas por mutações e recombinações são avaliadas e as melhores soluções são selecionadas para a nova geração de cromossomos, até que uma solução satisfatória seja construída. A avaliação das soluções geradas em cada fase da evolução é feita

considerando as restrições de interações entre os componentes da solução e os sub-problemas que cada componente atende, empregando uma codificação do conhecimento do domínio para as verificações.

### 3.4 Adaptação por Geração

As estratégias de adaptação por substituição e por transformação corrigem uma solução prévia para adequar-se a uma nova situação. Algumas vezes é mais apropriado construir um nova solução repetindo os mesmos passos encadeados de inferências e computações que foram executados para a construção de uma solução recuperada. Nessa estratégia, o sistema re-executa toda a seqüência de passos de um caso recuperado para construir a solução para um novo problema. Essa estratégia é útil quando cálculos intermediários devem ser feitos para derivar uma resposta a partir do enunciado do problema e quando os resultados desses cálculos intermediários são dependentes de valores das constantes do problema e os valores das constantes no problema atual e antigo não são as mesmas. Essas condições podem ocorrer em muitas situações de planejamento (Kolodner, 1993).

Para a aplicação dessa estratégias de adaptação, o sistema de RBC precisa implementar mecanismos que permitam que os casos armazenem mais do que apenas a solução prévia. Em particular, eles precisam armazenar as inferências ou computações que resultaram na solução prévia e as razões pelas quais essas inferências foram realizadas.

Em (Plaza e Arcos, 2002) é proposto uma forma de adaptação por geração, denominada adaptação construtiva. Nesse método de adaptação, a solução para um novo problema é construída - em contraste com os métodos de adaptação por transformação. Porém, esse método utiliza a solução do caso prévio recuperado e não um *trace* de raciocínio normalmente utilizado na adaptação por geração. O método proposto consiste de uma forma de algoritmo de busca heurística *best-first* que executa buscas no espaço das soluções utilizando informações dos casos prévios como guia. A adaptação construtiva trabalha com estados da solução para o problema atual. O algoritmo de busca heurística segue de um estado inicial, executando operações que levem a estados mais próximos do final, até a construção de uma solução completa para o novo problema.

Em (Carbonel, 1986) é proposto um método de adaptação de casos por geração que utiliza uma seqüência de ações e justificativas armazenadas como solução em casos prévios. Inicialmente, é apresentado ao sistema um problema consistindo de um modelo físico básico a ser simplificado. O sistema então recupera um caso prévio que possui um modelo básico com as mesmas



características, e uma seqüência de ações e decisões que devem ser executadas para simplificar o modelo básico inicial. O sistema extrai essa seqüência de ações e as executa para obter a solução para o novo problema.

### 3.5 Considerações Finais

Neste Capítulo foram apresentados as principais estratégias de adaptação de casos em sistemas de RBC propostas na literatura.

Foi visto que as estratégias de adaptação dividem-se, basicamente, em três grupos: adaptação por substituição, adaptação por transformação e adaptação por geração.

Um exame da literatura mostra que a maioria das propostas para adaptação de casos são específicas para o domínio da aplicação e raramente são reutilizáveis em outros domínios. Porém, algumas propostas visam uma certa independência do domínio, mas somente podem ser aplicadas a classes de problemas.

Em algumas aplicações, a utilização de abordagens híbridas que empregam RNAs ou Algoritmos Genéticos (AGs) mostraram resultados promissores na redução do esforço de aquisição de conhecimento para adaptação e na construção de métodos que possam ser utilizados em diferentes domínios de problemas.

A construção de abordagens de adaptação genéricas é tida como improvável (Kolodner, 1993), apesar de essa ainda ser uma área de pesquisa em aberto. As pesquisas atuais em adaptação de caso visam a redução do esforço de aquisição de conhecimento específico do domínio de aplicação.





---

## Sistemas Híbridos de RBC

---

### 4.1 *Considerações Iniciais*

Pesquisas sobre a integração de modelos híbridos têm apresentado um grande crescimento nos últimos anos, tanto na área de sistemas inteligentes, como na área de modelos cognitivos. As principais motivações para essas pesquisas são (Malek, 2001):

- processos cognitivos não são homogêneos e uma grande variedade de representações, mecanismos e técnicas precisam ser empregadas para modelá-los;
- o desenvolvimento de sistemas inteligentes para aplicações práticas pode se beneficiar da combinação de diferentes técnicas de IA, uma vez que nenhuma técnica sozinha pode solucionar todos os problemas.

Dentro da área de pesquisa de modelos híbridos, várias abordagens de integração de RBC com outras técnicas de IA foram propostas nos últimos anos (Reategui e Campbell, 1994; Malek e Amy, 1994; Reategui, 1995; Gómez et al., 1999; Puvis e Athalye, 1997; Jha et al., 1999; Malek, 2001; Sovat e Carvalho, 2001; Jarmulak et al., 2001; Craw, 2003; Pérez et al., 2002; Policastro et al., 2003a; Policastro et al., 2003b; Policastro et al., 2003c). Essa integração pode ser obtida pela divisão de tarefas entre o RBC e a outra técnica de IA ou pelo projeto de uma arquitetura inteligente combinando os dois paradigmas (Braga et al., 2000).

A maioria das abordagens propostas utiliza outras técnicas de IA para compor a fase de recuperação do ciclo de raciocínio de RBC. Outras propostas

utilizam outras técnicas de IA para a extração de conhecimento geral do domínio que é então utilizado durante o ciclo de raciocínio do RBC. A literatura apresenta também algumas propostas de utilização de outras técnicas de IA para a adaptação de casos.

Este capítulo apresenta uma visão geral das abordagens de integração neurosimbólica e simbólico-evolutiva. Na Seção 4.2 é apresentado um *framework* sobre a integração neurosimbólica. Na Seção 4.3 é feita uma breve discussão sobre a integração simbólico-evolutiva. Na Seção 4.4 é mostrado um breve estudo sobre sistemas híbridos de RBC. Na Seção 4.5 são apresentadas as considerações finais do capítulo.

## 4.2 Integração Neurosimbólica

De acordo com os fundamentos teóricos das abordagens neurais e simbólicas, duas categorias principais são distinguidas em sistemas neurosimbólicos clássicos, como mostrado na Figura 4.1 (Hilario, 1997):

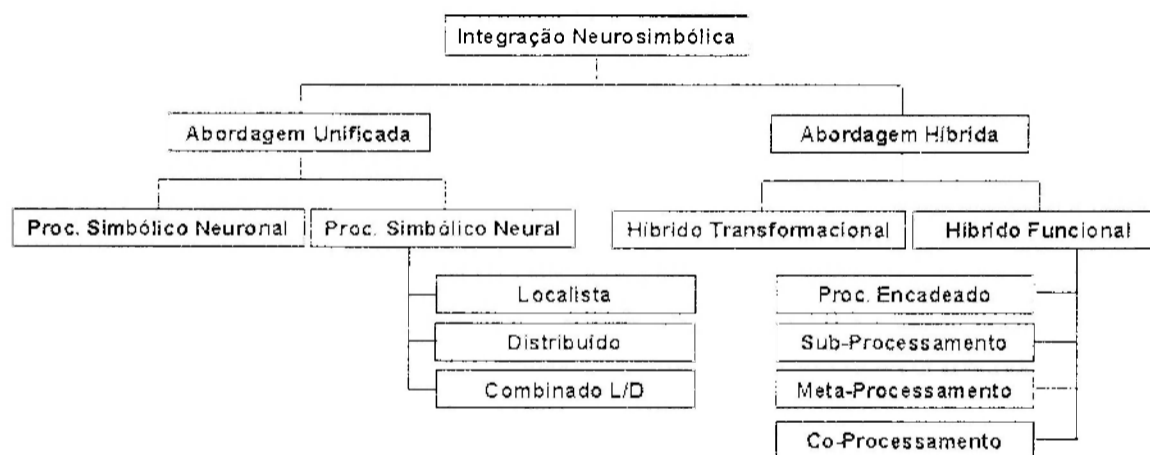


Figura 4.1: Estratégias de integração neurosimbólica (Hilario, 1997).

### 4.2.1 Abordagem Unificada

As alternativas da abordagem unificada são baseadas na premissa de que não existe a necessidade de processadores e estruturas de conhecimento simbólicos. O processamento simbólico emerge das estruturas e processos neurais. A proposta é atingir a capacidade neural e simbólica utilizando apenas uma RNA. As alternativas da estratégia unificada podem ainda ser divididas em duas tendências (Hilario, 1997): Processamento Simbólico *Neuronal*<sup>1</sup> e

<sup>1</sup>O termo *neuronal* denota uma identificação próxima aos aspectos fisiológicos do neurônio biológico.

## 4.2 Integração Neurosimbólica

---

Processamento Simbólico Neural<sup>2</sup>. A seguir, essas tendências serão discutidas brevemente.

### *Processamento Simbólico Neuronal*

É um caso especial de processamento *neuronal*. Seu objetivo é modelar as funções de alto nível do cérebro. Utiliza uma abordagem *bottom-up* e tem como ponto de partida o neurônio biológico.

### *Processamento Simbólico Neural*

É baseado na plausibilidade neurobiológica. O neurônio considerado aqui é geralmente um neurônio artificial formal. As RNAs são utilizadas para a construção de blocos capazes de realizar o processamento simbólico complexo.

## 4.2.2 Abordagem Híbrida

As alternativas da abordagem híbrida baseiam-se em suposições de que um processamento cognitivo completo e poderoso somente pode ser alcançado pela combinação de abordagens conexionistas e simbólicas.

De acordo com a sinergia da combinação dos modelos neural e simbólico, as estratégias híbridas podem ser divididas em duas classes (Hilario, 1997; Malek, 2001): híbridos funcionais e híbridos transformationais. A seguir, essas duas classes serão discutidas em mais detalhes.

### *Híbrido Funcional*

Incorporam componentes conexionistas e simbólicos completos. Em adição à RNA, ele contém uma estrutura simbólica com seu respectivo processamento.

Os modelos híbridos funcionais podem ser distinguidos por meio de várias dimensões, como o tipo de problema a que eles atendem, os modelos de componentes conexionista e simbólico utilizados, e o papel de cada componente no processamento do sistema.

As estratégias híbridas, podem ainda ser distinguidas de acordo com o grau de integração dos modelos neural e simbólico (Hilario, 1997):

- sistemas fracamente acoplados, nos quais os componentes têm uma clara localização no espaço e tempo de processamento. O controle e os dados podem ser transferidos diretamente entre os componentes neural e simbólico, ou por meio de um controle ou agente central, mas a interação é sempre iniciada por um dos componentes ou pelo agente.

---

<sup>2</sup>O termo *neural* denota uma similaridade com os aspectos funcionais do neurônio biológico.

- sistemas fortemente acoplados, nos quais os dados não são transferidos entre os componentes conexionista e simbólico, mas sim compartilhados por meio de uma estrutura interna comum.

As estratégias híbridas também possuem diferentes modos de integração entre os modelos neural e simbólico (Hilario, 1997):

- no processamento encadeado, um dos dois componentes (neural ou simbólico) executa o processamento principal e o outro utiliza o processamento executado para completar uma tarefa.
- no sub-processamento, um dos dois componentes (neural ou simbólico) é embutido e subordinado ao outro componente que executa as principais funções do processamento. Tipicamente, o componente neural é o componente que fica subordinado ao componente simbólico.
- no meta-processamento, um dos componentes (neural ou simbólico) executa a função de processamento do conhecimento, e o outro executa a função de meta-processamento (controle da melhoria do desempenho do sistema, por exemplo).
- no co-processamento, os dois componentes (neural e simbólico) são parceiros equivalentes no processamento e resolução de problemas. Eles podem interagir igualmente com o ambiente e enviar ou receber informações do outro componente. Eles podem trabalhar sob um supervisor ou meta-processador, cooperando para a resolução de um problema de diferentes maneiras, executando sub-tarefas distintas para compor um resultado ou executando a mesma tarefa sob condições diferentes e confrontando os resultados para a formação de uma resposta final do sistema.

#### *Híbrido Transformacional*

Podem ser vistos como uma classe entre os modelos unificados e os modelos híbridos funcionais. Os modelos transformacionais utilizam uma RNA como o centro do processamento, mas o processamento inicia ou termina em uma estrutura simbólica. As estruturas simbólicas não executam nenhum processamento, mas apenas servem como fonte ou destino do processamento da RNA. Essa abordagem pode ser vista como semi-híbrida, uma vez que utiliza estruturas simbólicas, mas não utiliza o processamento simbólico.

### 4.3 Integração Simbólico-Evolutiva

A integração de características evolutivas ao poder do processamento simbólico pode produzir sistemas híbridos robustos especialmente aplicáveis a problemas de otimização e busca, tendo despertado um grande interesse nos últimos anos.

Um exame da literatura e dos fundamentos teóricos das abordagens simbólica e evolutiva mostram três possíveis estratégias de integração entre esses dois paradigmas, como ilustrado na Figura 4.2:

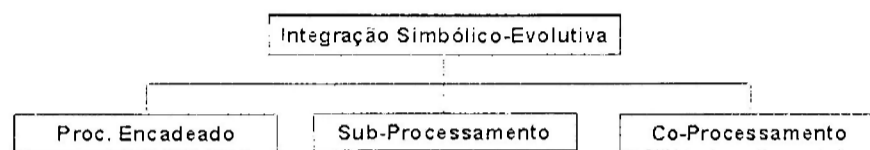


Figura 4.2: Estratégias de integração simbólico-evolutiva.

As estratégias de integração podem ainda ser distinguidas de acordo com o grau de integração dos modelos simbólico e evolutivo:

- sistemas fracamente acoplados, nos quais o controle e os dados podem ser transferidos diretamente entre os componentes simbólico e evolutivo por meio de um controle ou agente central, mas a interação é sempre iniciada pelo controle ou agente central, como em (Kaulfman e Michalski, 2000).
- sistemas fortemente acoplados, nos quais os dados são compartilhados por meio de uma estrutura interna comum, como em (Pérez et al., 2002)

#### *Processamento Encadeado*

No processamento encadeado, um dos dois componentes (simbólico ou evolutivo) executa o processamento principal e o outro utiliza o processamento executado para completar uma tarefa, como em (Araujo et al., 2000).

#### *Sub-processamento*

No sub-processamento, um dos dois componentes (simbólico ou evolutivo) é embutido e subordinado ao outro componente que executa as funções principais do processamento. Ambos os componentes (simbólico ou evolutivo) podem ser subordinados ao outro componente, como em (Gomez et al., 1999; Pérez et al., 2002).

### Co-processamento

No co-processamento, os dois componentes (simbólico e evolutivo) são parceiros iguais no processamento e resolução de problemas. Eles podem trabalhar sob um supervisor ou controle central, cooperando para a resolução de um problema de diferentes maneiras, executando sub-tarefas distintas para compor um resultado ou executando a mesma tarefa sob condições diferentes, como em (Kaulfman e Michalski, 2000; Carvalho e Freitas, 2000).

## 4.4 RBC Híbrido

As estratégias de integração de modelos híbridos propostas em (Hilario, 1997; Malek, 2001) podem ser generalizadas para um conjunto de estratégias de integração de RBC com outras técnicas de IA, como mostrado na Figura 4.3:

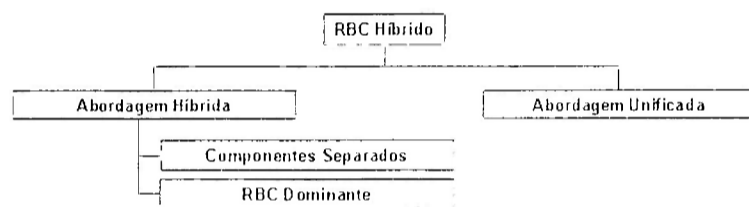


Figura 4.3: Estratégias de integração de sistemas híbridos de RBC.

A seguir, é apresentada uma breve discussão sobre cada uma dessas estratégias.

### 4.4.1 Abordagem Híbrida

Essa abordagem utiliza um módulo de RBC e outra técnica de IA, com diferentes modos de integração entre os modelos, para aprimorar o processo de resolução de problemas.

#### *Componentes Separados*

Nessa abordagem, o RBC e a outra técnica de IA são considerados como componentes distintos e completos que contribuem igualmente para a resolução de um problema. Três modos de integração podem ser distinguidos: co-processamento, meta-processamento e processamento encadeado.

Na literatura foram propostos vários sistemas híbridos com módulos de RBC e outras técnicas de IA como parceiros durante o processo de resolução de problemas. Esses sistemas geralmente utilizam um controle central



que alimenta os módulos de operação e obtém suas respostas para compor a resposta final do sistema.

Em (Alexandre et al., 1996) é proposta uma arquitetura genérica para sistemas híbridos para ser utilizada em diferentes aplicações. A abordagem proposta consiste em analisar a densidade dos dados utilizando clusterização e em utilizar uma integração apropriada entre um módulo de RBC e várias Redes Neurais Artificiais (RNAs).

Na arquitetura proposta, as RNAs trabalham com regiões de decisão onde a densidade dos dados é alta e o módulo de RBC trabalha com as regiões de baixa densidade - contendo casos atípicos.

Durante o processo de resolução de problemas, um controle central decide se o problema apresentado é um caso atípico ou não e alimenta o módulo apropriado (RBC ou RNA).

Algumas abordagens propostas utilizam, ainda, um módulo de raciocínio (RBC ou outra técnica de IA) como pré-processador de informações para o outro módulo.

Em (Reategui, 1995), uma RNA baseada no modelo neural combinatorial é utilizada para aquisição de conhecimento geral do domínio. Esse conhecimento adquirido pela rede é então interpretado e armazenado em estruturas de memória denominadas descritores de diagnóstico.

Cada descritor de diagnóstico representa e destaca as mais importantes características para a identificação de um diagnóstico, e para a associação do diagnóstico com experiências prévias relevantes.

O conhecimento extraído da RNA consiste de três tipos de características:

1. os *trigger*: características ou combinações de características muito específicas e freqüentes nos respectivos diagnósticos que elas identificam;
2. as *características primárias*: características ou combinações de características que podem ser específicas ou freqüentes nos respectivos diagnósticos que elas identificam e conferem um alto grau de confiança na identificação de uma hipótese de diagnóstico;
3. as *características de suporte*: características ou combinações de características que não são específicas nem freqüentes nos respectivos diagnósticos, mas podem reforçar as conclusões de um diagnóstico.

O conhecimento extraído da RNA também é composto por dois tipos de características adicionais, que podem ser utilizados como índices positivos e negativos da BC quando o conhecimento do domínio não for suficiente.

Em (Krovvidy e Wee, 1992) é proposto um sistema para o tratamento de água (denominado WATTS), que integra várias técnicas de IA.

O sistema proposto utiliza o algoritmo indutivo ID3 para extrair conhecimento em forma de regras de uma base de dados de seqüências de tratamentos. Essas regras são então utilizadas pelo seqüenciador de processos para obter uma seqüência de tratamentos para um problema apresentado.

O seqüenciador de processos foi desenvolvido utilizando duas abordagens: uma utiliza um algoritmo de busca heurística baseado no algoritmo A\*, a outra utiliza uma RNA. Uma RNA baseada no modelo de Hopfield é utilizada para resolver o problema de otimização da solução que está sendo criada, enquanto o algoritmo de busca heurística é utilizado para gerar uma solução ótima.

No sistema proposto, um módulo de RBC é utilizado para lembrar seqüências prévias de tratamento e soluções iniciais para a RNA ou para o algoritmo de busca heurística.

Em (Prentzas e Hatzilygeroudis, 2002) é apresentada uma abordagem que integra raciocínio baseado em casos e raciocínio baseado em neuro-regras. Neuro-regra é um tipo de regra híbrida que integra regra simbólica com computação neural.

Cada neuro-regra é implementada por uma unidade de processamento do tipo *Adaline*. As entradas da Adaline são as condições da neuro-regra. Os pesos das conexões são os fatores de importância das condições e existe ainda um fator de *bias* para cada neuro-regra. Cada neuro-regra pode produzir a saída: 1 (verdadeiro), -1 (falso) ou 0 (desconhecido). As neuro-regras podem ser construídas a partir de regras simbólicas. As regras simbólicas que levam à mesma conclusão são mescladas e uma unidade Adaline é associada a cada regra. Em seguida, cada unidade Adaline é treinada individualmente por um algoritmo *Least Mean Square*<sup>3</sup>.

O sistema é composto por uma memória de trabalho, um mecanismo híbrido de inferência, um mecanismo de explicações, uma base de neuro-regras e uma BC. A BC contém casos que constituem exceções às neuro-regras e são indexados por essas. O mecanismo de inferência combina o raciocínio baseado em neuro-regras com o raciocínio baseado em casos. Durante o raciocínio baseado em regras, quando existem condições suficientes para disparar uma neuro-regra, o mecanismo de inferências verifica se existem casos que constituem exceções à neuro-regra sendo disparada. Caso não exista, o mecanismo de inferência dispara a neuro-regra e continua o processo de raciocínio. caso existam casos associados, o mecanismo de inferência produz uma explicação, realiza o raciocínio baseado em casos e efetua um teste para concluir se o processo de raciocínio utilizará o resultado do caso recuperado ou da neuro-regra.

---

<sup>3</sup>Esse termo é utilizado em inglês por ser amplamente difundido na comunidade.

#### *RBC Dominante*

Nessa estratégia, a outra técnica de IA é utilizada em uma ou mais fases do Ciclo de RBC, correspondendo à abordagem de sub-processamento.

A maioria dos abordagens propostas de integração de sistemas híbridos de RBC dominante utilizam uma outra técnica de IA para compor alguma parte do ciclo de RBC, em especial a indexação e o processo de recuperação de casos prévios.

Em (Craw, 2003), o algoritmo C4.5 é utilizado para a extração de conhecimento que é então utilizado para a recuperação de casos. O conhecimento é extraído da própria case de casos.

Em (Malek e Amy, 1994) uma RNA incremental baseada em protótipos é utilizada para indexação e recuperação de casos. Essa rede é composta por duas <sup>4</sup> camadas: as unidades de entrada apresentam à rede cada atributo do problema, a camada de saída contém um neurônio para cada classe, a camada interna contém os neurônios para cada protótipo dos casos prévios armazenados.

A idéia é construir um sistema simples de indexação que contém dois níveis de memória: a memória que contém os protótipos dos casos e a memória que contém as instâncias dos casos reais.

A memória de protótipos é utilizada como um sistema de indexação durante a fase de recuperação para reduzir o tempo do processo. O primeiro nível de memória é a camada escondida da rede baseada em protótipo. O segundo nível de memória é uma memória plana que mantém as instâncias dos casos em grupos de similaridade. Cada protótipo do primeiro nível é ligado a um grupo de casos do segundo nível, contendo todas as instâncias de casos pertencentes a esse protótipo. Além disso, existe um outro agrupamento que contém os casos que caem nas regiões de incerteza de classificação da RNA, bem como os casos que não foram classificados. Esses casos são chamados de casos atípicos.

Em (Main e Dillon, 1999; Main e Dillon, 2001) foi proposto um sistema de RBC que utiliza várias RNAs do tipo *Perceptron* Multi - Camadas (MLP) (Braga et al., 2000) para indexação e recuperação dos casos. Os casos são organizados em uma representação orientada a objetos onde cada nível da hierarquia de classes representa um refinamento das classes de casos. O sistema utiliza uma RNA de duas camadas (uma camada intermediária e uma de saída) para controlar a indexação e recuperação dos casos em cada nível da hierarquia de classes.

Quando os casos são indexados no sistema, o seu rótulo deve ser pré-

---

<sup>4</sup>nesse trabalho a convenção adotada é a de que os neurônios de entrada não representam uma camada de nós

determinado. Dessa forma, o sistema utiliza treinamento supervisionado para adicionar o novo caso à estrutura de indexação. Durante o processo de recuperação, cada rede é utilizada para determinar a sub-classe à qual o problema apresentado pertence. Esse processo continua até que se atinja o nível mais baixo da hierarquia de classes e um caso real seja recuperado.

Algumas propostas também utilizam RNAs incorporadas ao ciclo de RBC durante o processo de adaptação dos casos prévios recuperados.

Em (Corchado et al., 1998; Corchado et al., 2001), conforme apresentado no Capítulo 3, Subseção 3.2, uma RNA baseada no modelo RBF (Orr, 1996) é utilizada durante a fase de adaptação de casos. Quando um novo problema é apresentado ao sistema, um número de casos prévios similares é recuperado e esses casos são utilizados para treinar uma rede RBF, que é então utilizada para propor a solução final para o problema atual.

Em (Wiratunga et al., 2002), conforme apresentado no Capítulo 3, Subseção 3.2, um comitê de algoritmos Rise (Domingos, 1996) - é criado utilizando a técnica de boosting (Freund e Schapire, 1996) e utilizado para a adaptação da solução de um caso recuperado.

Em (Lotfy e Sameh, 2002) uma RNA baseada no modelo MLP é utilizada durante a fase de adaptação de casos. Quando um novo problema é apresentado ao sistema, um número de casos prévios similares é recuperado e esses casos são utilizados para treinar uma rede MLP composta por três <sup>5</sup> camadas: são utilizados onze neurônios de entrada, duas camadas intermediárias contendo cinco e quatro neurônios respectivamente e uma camada de saída contendo um neurônio. Após o treinamento da RNA com o conjunto de casos recuperados, essa torna-se capaz de aproximar uma função que pode interpolar novos valores baseados nos valores dos casos recuperados. Finalizado o treinamento, a descrição do novo problema é apresentada para a rede e essa fornece a solução para o problema atual.

Algumas abordagens propostas utilizam um AG durante o processo de adaptação da solução do caso recuperado.

Em (Gómez et al., 1999; Puvis e Athalye, 1997; Garza e Maher, 2000) um AG é utilizado para a adaptação de casos em um sistema de RBC. Nessas abordagens, os casos são adaptados de maneira incremental e em paralelo até que uma solução satisfatória para o novo problema seja construída. Esse método de adaptação suporta dois tipos de adaptação: paramétrica e estrutural. A adaptação paramétrica é alcançada por mutação. A adaptação estrutural é alcançada por meio de cruzamento.

Quando um novo problema é apresentado ao sistema, vários casos satis-

---

<sup>5</sup>nesse trabalho a convenção adotada é a de que os neurônios de entrada não representam uma camada de nós

fazendo parcialmente a solução do problema são recuperados. Esses casos recuperados alimentam o AG, ou seja, formam a população inicial de cromossomos.

As adaptações das soluções produzidas por mutações e cruzamentos são avaliadas e as melhores soluções são selecionadas para a nova geração de cromossomos, até que uma solução satisfatória seja obtida.

A avaliação das soluções geradas em cada fase da evolução é feita considerando as restrições de interações entre os componentes da solução e os sub-problemas que cada componente atende, utilizando uma codificação do conhecimento do domínio para as verificações.

Existem ainda propostas para a utilização de RNA no processo para controlar o processo de manutenção da BC dos sistemas de RBC.

Em (Shiu et al., 2001) foi proposta uma abordagem para a manutenção de BCs utilizando uma RNA de duas camadas (uma camada intermediária e uma de saída) para auxiliar na classificação e seleção dos casos representativos da base. A abordagem híbrida de manutenção é dividida em três fases.

Na primeira fase, o sistema utiliza a RNA para agrupar os casos da BC e determinar uma métrica de relacionamento que define o quanto cada caso pertence a um dado *cluster*.

Na segunda fase, o conjunto de *clusters* é refinado através de uma transformação que pode ser feita utilizando gradiente descendente. Essa fase pode ser ignorada de acordo com a classificação inicial da rede. Caso a RNA consiga agrupar um número satisfatório de casos da base, a fase de refino do conjunto de *clusters* pode ser ignorada.

A última fase utiliza o conjunto de *clusters* e a métrica de relacionamento para formular a política de exclusão de casos da base.

A política de exclusão determina a densidade dos *clusters* e quais casos são mais representativos dentro desses *clusters*. Os casos menos representativos podem ser excluídos se necessário.

#### 4.4.2 Abordagem Unificada

Essa estratégia utiliza uma implementação conexionista de um sistema de RBC, onde uma RNA é utilizada para produzir um sistema de RBC para resolver um determinado problema. Algumas implementações conexionistas de sistemas de RBC foram propostas em (Becker e Jazayeri, 1989a; Becker e Jazayeri, 1989b; Myllimaki e Tirri, 1994). Nessas abordagens, diferentes tipos de neurônios (entrada, internos e de saída) são utilizados para desempenhar os papéis necessários às tarefas do sistema de RBC. Os neurônios de entrada, em geral, recebem os valores dos atributos da descrição do problema. Existe um neurônio de entrada para cada atributos da descrição do problema.

Em algumas abordagens, os neurônios internos formam uma representação dos casos armazenados na memória do sistema. Em outras abordagens, os neurônios internos auxiliam a adaptar e a produzir a solução para um novo problema. Os neurônios de saída produzem a solução para um novo problema apresentado.

#### 4.5 Considerações Finais

Neste Capítulo foram apresentados as principais abordagens de integração de sistemas de RBC com RNAs e AGs e outros algoritmos de AM. Foi apresentado também um *framework* sobre a integração neurosimbólica, uma área de grande interesse da comunidade de pesquisadores.

Na maioria das propostas de integração de algoritmos de AM com RBC, os algoritmos têm sido utilizadas para a indexação e recuperação de casos. Outras propostas utilizam algoritmos de AM para extrair o conhecimento geral do domínio para apoiar o processo de raciocínio. Outras ainda utilizam algoritmos de AM durante o processo de adaptação da solução do caso recuperado.

Neste capítulo também foi mostrada uma estrutura geral de integração simbólico-evolutiva, uma área de pesquisa relativamente nova e de grande interesse.

Nesse ponto, deve ser ressaltado mais uma vez que o ciclo RBC se propõem a fornecer uma metodologia consistente de raciocínio e aprendizado. A utilização de outras técnicas de IA, para compor o ciclo RBC pode fornecer sistemas inteligentes mais eficientes em diversas aplicações práticas.

---

## Algoritmos e Métodos Utilizados

---

### 5.1 *Considerações Iniciais*

Para a construção das abordagens híbridas de adaptação de casos investigadas nesta Dissertação de Mestrado foram estudados vários conceitos e técnicas de AM, bem como algumas técnicas para o pré-processamento dos conjuntos de dados utilizados na avaliação das abordagens construídas.

Na Seção 5.2 são apresentados alguns conceitos de AM. Na Seção 5.3 são mostradas as técnicas de pré-processamento utilizadas durante os experimentos realizados. Na Seção 5.4 são descritos os algoritmos utilizados nos experimentos realizados nesta Dissertação de Mestrado. Na Seção 5.5 são discutidos os métodos utilizados para a análise dos resultados obtidos. Na Seção 5.6 são apresentadas as considerações finais do capítulo.

### 5.2 *Aprendizado de Máquina*

AM é uma área da IA cujo objetivo é o estudo de técnicas computacionais de aprendizado que permitam desenvolvimento de sistemas capazes de adquirir conhecimento de forma automática. Um algoritmo de aprendizado é um programa computacional capaz de tomar decisões baseadas em soluções de problemas anteriores. Os sistemas de aprendizado possuem características que permitem sua classificação de acordo com a linguagem de descrição, modo, paradigma e forma de aprendizado. Na Tabela 5.1 são mostradas algumas dessas características de classificação dos sistemas de aprendizado.

Tabela 5.1: Características Gerais dos Sistemas de Aprendizado (Monard e Baranauskas, 2003).

Modos de Aprendizado	Paradigmas de Aprendizado	Linguagens de Descrição	Formas de Aprendizado
Supervisionado	Simbólico	Exemplos ou Objetos	Incremental
Não-Supervisionado	Estatístico	Hipóteses	Não-Incremental
	Baseado em Instâncias	Conhecimento do Domínio	
	Conexionista		
	Evolutivo		

### 5.2.1 Aprendizado Indutivo

Indução é uma forma de inferência lógica. Ela permite que conclusões sejam tomadas a partir de conhecimento particular. É uma forma de raciocínio que parte do específico para o geral, da parte para o todo (Batista, 2003). Hipóteses geradas a partir de inferências indutivas podem não preservar a verdade, levando a conclusões que excedam o conteúdo das premissas ou a conclusões falsas.

Um tipo de aprendizado indutivo é o aprendizado por exemplos. Nesse tipo de aprendizado, são induzidos conceitos gerais utilizando exemplos específicos desses conceitos (Michalski et al., 1983).

Por meio de generalizações e especificações, é possível criar um algoritmo que induz uma hipótese de um conceito consistente com todos os exemplos desse conceito. Esse Algoritmo inicia induzindo uma hipótese consistente para um único exemplo positivo e adiciona novos exemplos procurando manter a consistência da hipótese (Batista, 2003).

Existem muitos desafios em aprender um conceito a partir de exemplos. Deve-se ter cuidado ao induzir uma hipótese para que ela não se torne especializada para os exemplos utilizados para a indução, impossibilitando a generalização do conceito e o reconhecimento desse conceito em novos exemplos apresentados ao algoritmo após o processo de indução e criação da hipótese. Outro desafio é o aprendizado de um conceito quando seus exemplos possuem ruído. A existência de ruído pode criar uma hipótese consistente com valores incorretos. Nessas situações, é preferível induzir uma hipótese mais simples e não consistente com todos os exemplos (Batista, 2003).

### 5.3 Pré-processamento de Dados

Freqüentemente, os conjuntos de dados coletados para serem utilizados como base de conhecimento apresentam diversos problemas, como valores desconhecidos, desproporção entre o número de exemplos de cada conceito



contido no conjunto de dados (dados não balanceados), entre outros. Para tentar aprimorar a qualidade dos dados coletados antes de utilizá-los, algumas técnicas de pré-processamento podem ser empregadas e aplicadas sobre o conjunto de dados. Nesta Seção serão discutidas brevemente as técnicas de pré-processamento empregadas neste trabalho de Mestrado.

#### 5.3.1 Tratamento de Valores Desconhecidos

Um problema em pré-processamento de dados é o tratamento de valores desconhecidos (Batista, 2003). Valores ausentes consistem na não medição de valores para um ou mais atributos em determinadas situações como recusa por parte de entrevistados em responder determinadas perguntas, defeitos em equipamentos, entre outras.

A aleatoriedade dos valores desconhecidos é um fator importante a ser analisado para a escolha do método de tratamento dos valores desconhecidos (Batista, 2003). Na sua forma mais simples, os valores desconhecidos podem estar distribuídos aleatoriamente nos dados. Porém, valores desconhecidos podem não estar aleatoriamente distribuídos. Isso significa que a propabilidade de se encontrar um valor desconhecido pode estar ligado ao valor verdadeiro do valor desconhecido ou a valores dos demais atributos do conjunto de dados.

Muitas técnicas têm sido aplicadas, algumas delas são bastante simples, como a substituição do valor desconhecido pela média ou pela moda do atributo. A técnica utilizada neste trabalho para o preenchimento de valores ausentes é proposta em (Batista, 2003). Essa técnica utiliza um algoritmo *k-Vizinhos mais Próximos* para obter a média ou a moda dos *k*-vizinhos mais próximos do exemplo que contem o atributo com o valor ausente. Para o pré-processamento foi utilizado o valor de *k* igual a 3, pois esse valor possibilitou a obtenção de melhores resultados. Portanto, para cada exemplo contendo um atributo com valor desconhecido, 3 exemplos foram recuperados para o cálculo da média ou da moda para o atributo. Em seguida, o valor encontrado foi utilizado para substituir o valor desconhecido.

#### 5.3.2 Tratamento de Classes Desbalanceadas

O problema de classes desbalanceadas é de grande importância, uma vez que conjuntos de dados com essas características podem ser encontrados em vários domínios.

Muitos algoritmos de aprendizado assumem que suas classes estão balanceadas e falham em induzir um classificador que possa prever as classes minoritárias (Batista, 2003). O problema agrava-se quando o custo da clas-

sificação incorreta da classe minoritária é maior que o custo da classificação incorreta da classe majoritária. No mundo real, essa é a norma para a maioria dos domínios com conjunto de dados desbalanceados.

Para o tratamento de conjunto de dados com classes desbalanceadas foi utilizada uma técnica de over-sampling proposta em (Chawla et al., 2002). Essa técnica cria exemplos sintéticos para a classe minoritária, utilizando um algoritmo *k-Vizinhos mais Próximos*. Ela utiliza o algoritmo *k-Vizinhos mais Próximos* para recuperar exemplos similares a cada exemplo da classe minoritária. Cada exemplo recuperado pelo algoritmo *k-Vizinhos mais Próximos* é combinado com o exemplo da classe minoritária apresentado ao algoritmo de balanceamento para gerar um exemplo sintético. Para atributos contínuos, essa combinação é feita, atributo a atributo, por meio da extração das diferenças dos valores de cada atributo multiplicada por um número aleatório no intervalo 0..1. Esse processo de combinação permite a seleção aleatória de um valor ao longo do segmento entre os valores das características do atributo dos dois exemplos sendo combinados (Chawla et al., 2002). O exemplos sintéticos possibilitam que o classificador crie regiões de decisão maiores e menos específicas. Para atributos discretos, o algoritmo realiza uma escolha aleatória entre os valores encontrados para o atributo dos dois exemplos sendo combinados.

## 5.4 Algoritmos

Nesta Seção são discutidos brevemente os algoritmos utilizados nos experimentos realizados nesta Dissertação de Mestrado.

### 5.4.1 Rede ART2

As RNAs baseadas no modelo Teoria de Ressonância Adaptativa (ART2) pertencem a uma classe de RNAs que empregam aprendizado não supervisionado e que suportam o aprendizado contínuo, respondendo a mudanças dos padrões de entrada. Esse modelo de RNA aceita entradas com valores contínuos ou binários e é composta por três camadas ou estágios de processamento (Carpenter e Grossberg, 1987).

A camada  $F_0$  executa o pré-processamento e a normalização do padrão de entrada. Para fazer isto, esta camada utiliza uma função não-linear dependente do ruído das informações processadas. Esta função pode ser sigmoideal ou linear threshold (Carpenter e Grossberg, 1987).

A camada  $F_1$  processa o padrão filtrado (pela camada  $F_0$ ) e compara com os protótipos dos agrupamentos (armazenados na camada  $F_2$ ), por meio de um conjunto de pesos  $W$  (pesos *bottom-up*). Dado um padrão de entrada

pré-processado, uma unidade de processamento em  $F_2$  é selecionada. Essa unidade de processamento representa uma categoria (protótipo de um agrupamento). Uma vez selecionada uma unidade de processamento, ela envia seu protótipo para a camada  $F_1$  por meio de um conjunto de pesos  $T$  (pesos *top-down*). A rede ART2 efetua a comparação entre o protótipo do agrupamento selecionado e o padrão de entrada pré-processado. Se eles não são semelhantes o suficiente (dado um parâmetro de vigilância), o subsistema de orientação ativa um sinal de *reset* e desabilita a unidade de processamento selecionada em  $F_2$ . Como resultado, um novo ciclo de busca é iniciado. Se o protótipo do agrupamento selecionado e o padrão de entrada são semelhantes, os pesos do agrupamento selecionado são atualizados (Carpenter e Grossberg, 1987).

A função da camada  $F_2$  é efetuar a busca por um agrupamento (unidade de processamento) que seja mais similar ao padrão de entrada pré-processado. Esta busca é executada em paralelo. Todas as unidades entram em competição e a de maior ativação é a vencedora. Se a camada  $F_2$  recebe um sinal de *reset* do subsistema de orientação, a unidade vencedora da competição é desabilitada e ela não pode mais entrar em competição nas fases de busca subsequentes, para o padrão de entrada atual. As unidades de processamento restantes competem novamente, e o ciclo se repete até uma unidade ser encontrada para acomodar o novo padrão de entrada, ou uma nova unidade ser criada pela rede (Carpenter e Grossberg, 1987).

Externamente às camadas de processamento da rede ( $F_0$ ,  $F_1$ , e  $F_2$ ), a rede ART2 possui um subsistema de orientação que estabelece o grau de similaridade entre os protótipos dos agrupamentos e o padrão de entrada pré-processado. A condição de *reset* é dada por um parâmetro de vigilância cujo valor deve estar entre 0 e 1 ( $0 < \rho < 1$ ) (Carpenter e Grossberg, 1987). A arquitetura da rede ART2 é ilustrada na Figura 5.1.

#### 5.4.2 Algoritmo C4.5

O algoritmo C4.5 é um algoritmo de aprendizado simbólico que gera classificadores na forma de árvores de decisão e conjuntos de regras. Esse algoritmo constrói a árvore de decisão a partir de um conjunto de dados de treinamento pela aplicação da estratégia de dividir e conquistar e um algoritmo *greedy* que utiliza uma relação de ganho como critério para a construção da árvore. Ele seleciona um atributo de entrada, divide as instâncias de treinamento em sub-conjuntos correspondentes aos valores do atributo selecionado e calcula a relação de ganho para esse atributo. Esse processo é repetido para cada atributo de entrada das instâncias de treinamento. O C4.5 seleciona o atributo com melhor relação de ganho como raiz da árvore de decisão. O algoritmo é aplicado recursivamente para formar as sub-árvores do modelo, terminando

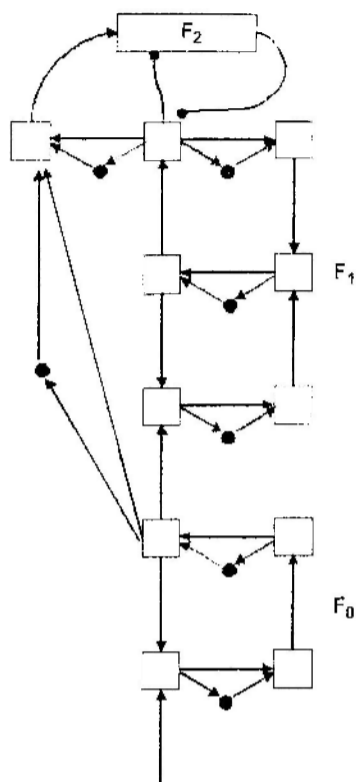


Figura 5.1: Arquitetura da rede ART2 (Carpenter e Grossberg, 1987).

quando um nó contém somente o atributo classe (Quinlan, 1993).

Após a construção da árvore de decisão, o modelo resultante é muitas vezes muito complexo e específico para o conjunto de dados de treinamento. Para melhorar a precisão e tornar a árvore inicial menos específica ao conjunto de dados de treinamento, o algoritmo C4.5 efetua um processo de poda da árvore, que consiste em retirar os nós que não contribuem para a predição de novos valores, produzindo uma árvore de decisão menos complexa e com uma melhor generalização.

Um exemplo da arquitetura de uma árvore de decisão gerada pelo algoritmo C4.5 a partir do conjunto de dados de pontes de Pittsburgh é ilustrada na Figura 5.2.

### 5.4.3 Algoritmo M5

O Algoritmo M5 é um algoritmo de aprendizado que gera modelos na forma de Árvores de Regressão (*Model Tree*) (Wang e Witten, 1997) combinada com equações de regressão. Esse tipo de árvore funciona da mesma maneira que uma árvore de classificação, porém os nós folha contêm expressões lineares, ao invés de valores das classes.

O algoritmo M5 constrói um *Model Tree* a partir de um conjunto de dados de treinamento, de uma maneira semelhante à geração de uma árvore de clas-

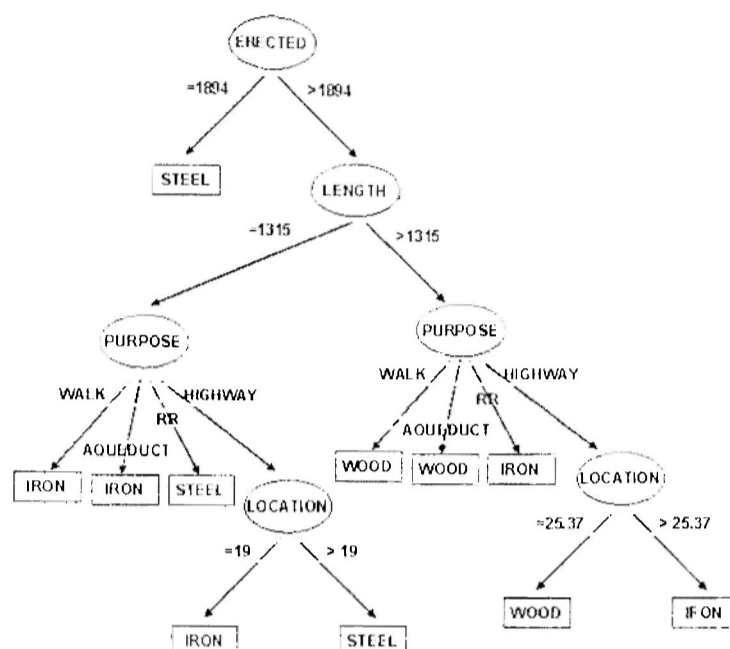


Figura 5.2: Exemplo de arquitetura de uma árvore de decisão.

sificação. Esse algoritmo constrói o *Model Tree* pela aplicação da estratégia de dividir e conquistar que utiliza uma relação de ganho pela minimização do desvio padrão dos valores de saída do conjunto de treinamento como critério para a construção do modelo. Ele seleciona um atributo de entrada, divide as instâncias de treinamento em sub-conjuntos correspondentes aos valores do atributo selecionado e calcula o desvio padrão dos valores de saída do conjunto de treinamento para esse atributo. Esse processo é repetido para cada atributo de entrada das instâncias de treinamento. O algoritmo seleciona o atributo com melhor relação de ganho como raiz da árvore de decisão. O algoritmo é aplicado recursivamente para formar as sub-árvores do modelo, terminando quando um dado contém somente o atributo classe (Quinlan, 1992).

Após a construção da árvore inicial, um modelo linear é calculado para cada exemplo nos nós interno da árvore, utilizando um processo padrão de regressão. Porém, ao invés de utilizar todos os atributos dos exemplos, apenas são utilizados os atributos que são testados nas sub-árvores pertencente ao nó corrente (Quinlan, 1992). Após cada modelo linear ser obtido, ele é simplificado pela eliminação de parâmetros para minimizar seu erro estimado. O algoritmo M5 utiliza um algoritmo de busca *greedy* para eliminar parâmetros que contribuem pouco com o modelo (Quinlan, 1992).

Posteriormente, alguns testes são realizados para podar a árvore em um processo que retrocede a partir de seus nós folha. Cada nó não terminal do *Model Tree* é examinado, iniciando dos nós próximo aos nós folha. O algoritmo M5 pode selecionar como modelo final para cada nó testado, tanto o modelo

linear simplificado calculado anteriormente, como o modelo das sub-árvores do nó sendo testado, dependendo de qual retorna um erro estimado mais baixo. Se o modelo selecionado para um nó é o linear, então as sub-árvores desse nó são eliminadas (Quinlan, 1992). Um exemplo da arquitetura do *Model Tree* gerado para a base de dados de servo mecanismo é ilustrada na Figura 5.3.

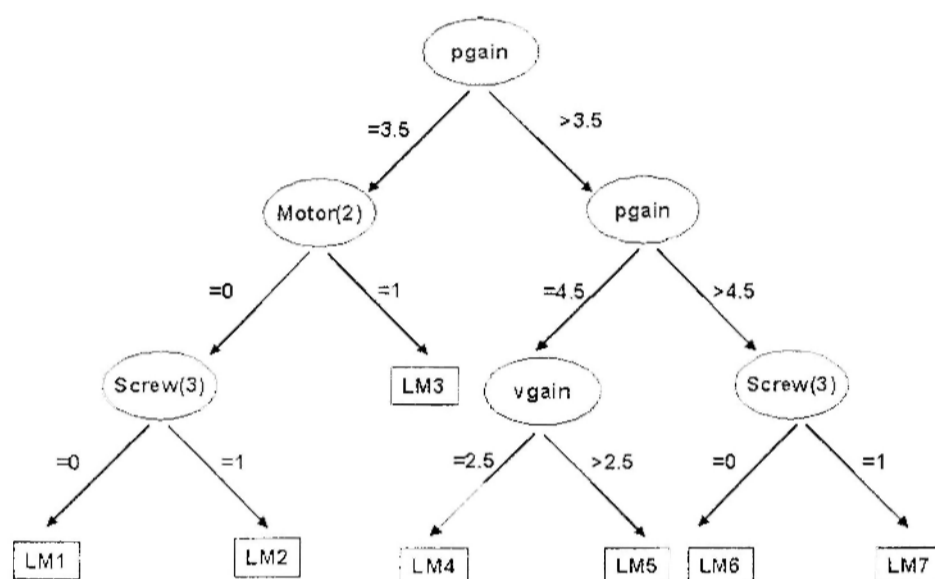


Figura 5.3: Exemplo de arquitetura do *Model Tree* (Wang e Witten, 1997).

#### 5.4.4 Rede MLP

As RNAs baseadas no modelo MLP podem resolver problemas complexos. Possuem uma ou mais camadas intermediárias com funções não-lineares (geralmente Sigmóide) que efetuam sucessivas transformações não lineares. As camadas intermediárias transformam problemas não linearmente separáveis em problemas linearmente separáveis e simplificam o problema para a camada de saída (Braga et al., 2000).

Uma rede MLP é uma rede que possui uma estrutura dividida em várias camadas. Cada camada contém 1 ou mais unidades de processamento que recebem suas entradas da camada imediatamente anterior e envia suas saídas para a camada imediatamente posterior. Não existem conexões entre as unidades de processamento de uma mesma camada. A camada de entrada é uma camada de simples recepção, não executando nenhum processamento. Esta camada alimenta as unidades de processamento da primeira camada escondida. A ativação das unidades de processamento  $k$  é uma função  $S_k$  das entradas ponderadas somado de um termo de limiar ( $\Theta$ ), dada pela Equação 5.1:

$$S_k(t) = \sum_{j=1}^n w_{jk}(t)g_j(t) + \theta_k(t) \quad (5.1)$$

A saída das unidades de processamento é distribuída para a camada escondidas seguintes até que a última camada da rede seja alcançada.

O algoritmo mais utilizado para o treinamento de redes MLP é o algoritmo *Back-Propagation*. Um algoritmo supervisionado que utiliza pares (entrada - saída) para ajustar os pesos das conexões da rede (Braga et al., 2000). A aplicação do algoritmo *Back-Propagation* envolve duas fases:

1. durante a primeira fase, as entradas  $X$  são fornecidas para a rede e seus sinais são propagados até as unidades de processamento de saída, onde as ativações são medidas e comparadas com as saídas necessárias. Esta comparação resulta em um erro para cada unidade de processamento de saída;
2. na segunda fase, os erros das unidades de processamento de saída são propagados para trás para a determinação do erro das unidades de processamento das camadas internas e efetuar os ajustes apropriados dos pesos das conexões.

A arquitetura da rede MLP é ilustrada na Figura 5.4.

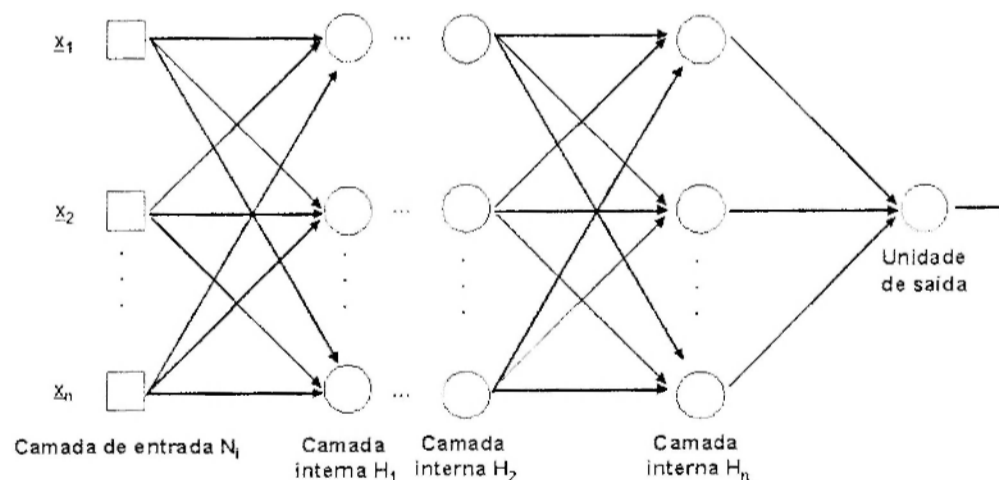


Figura 5.4: Arquitetura da rede MLP (Braga et al., 2000).

#### 5.4.5 Algoritmo SVM

Máquina de Vetores de Suporte (SVM) engloba uma classe de algoritmos de aprendizado baseado na teoria de aprendizado estatístico, combinando controle de generalização com uma técnica para tratar o problema da dimensio-

nalidade <sup>1</sup> (Vapnik, 1998). Assim como redes RBF e MLP, SVM pode ser utilizado para classificação de padrões ou para regressões não lineares (Haykin, 1999).

Basicamente, SVM é um algoritmo linear que constrói hiperplanos como superfícies de decisão de maneira que a fronteira de separação entre classes positivas e negativas seja maximizada (Haykin, 1999). Para isso, SVM segue um princípio com respaldo da Teoria do Aprendizado Estatístico: a minimização do risco estrutural. Esse princípio da indução é baseado no fato da taxa de erro do algoritmo de aprendizado sobre os dados de teste ser limitada pela soma da taxa de erro de treinamento e um termo que depende da dimensão *Vapnic-Chevonenkis* (VC). A dimensão VC é uma medida da capacidade ou poder de expressão da família de funções de classificação gerada por um algoritmo de aprendizado. Quanto maior seu valor, mais complexo o conjunto de funções de classificação (Haykin, 1999).

A noção central na construção de algoritmos SVM é o núcleo de produtos escalares entre um vetor de suporte  $x_i$  e um vetor  $x$  traçado no espaço da entrada (Haykin, 1999). Os vetores de suporte consistem de um pequeno subconjunto dos dados de treinamento extraídos pelo algoritmo. Dependendo de como o núcleo de produtos escalares é gerado, pode-se construir diferentes algoritmos de aprendizado caracterizados por superfícies de decisão não-lineares (Haykin, 1999). Em particular, pode-se utilizar Máquinas de Vetores de Suporte (SVMs) para a construção dos seguintes algoritmos de aprendizado (entre outros):

- algoritmos de aprendizado polinomiais;
- redes RBF;
- redes *Perceptron* de duas camadas;

Portanto, pode-se utilizar o algoritmo SVM para implementar o processo de aprendizado dessas RNAs, onde o número de unidades de processamento da camada interna da rede é automaticamente determinado a partir do conjunto de dados de treinamento (Haykin, 1999). A arquitetura de um classificador baseado na técnica SVM é ilustrada na Figura 5.5.

---

<sup>1</sup>Algoritmos de Aprendizado de Máquina podem ter seu desempenho prejudicado ao trabalhar com conjunto de dados com um número elevado de atributos (alta dimensionalidade), necessitando da utilização de técnicas de seleção de atributos para a redução da dimensionalidade original dos dados. SVM é capaz de manter uma boa generalização mesmo em dados com alta dimensionalidade.



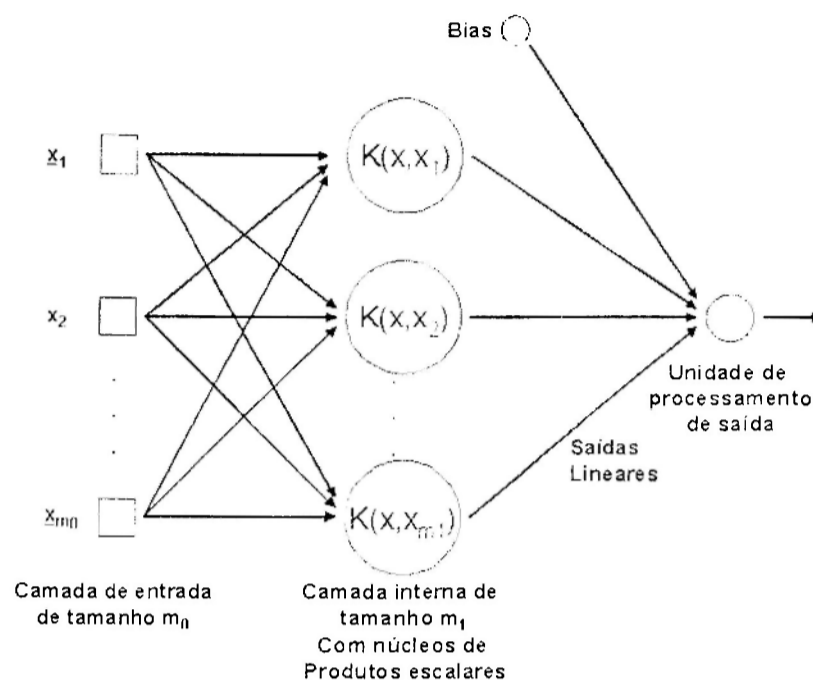


Figura 5.5: Arquitetura de um classificador SVM (Haykin, 1999).

#### 5.4.6 Algoritmo *K-Vizinhos mais Próximos*

O algoritmo *K-Vizinhos mais Próximos* é o mais básico método de aprendizado pertencente a uma família de métodos conhecidos como baseados em instância (Mitchell, 1997b). Os métodos de aprendizado baseados em instância são abordagens conceitualmente simples. O aprendizado nesses métodos consiste no simples armazenamento das instâncias de treinamento em uma memória. Quando uma nova instância é apresentada ao sistema, um conjunto de instâncias similares é recuperado da memória do sistema e utilizado para classificar a nova instância.

A versão básica do algoritmo *K-Vizinhos mais Próximos* assume que todas as instâncias correspondem a pontos em um espaço  $n$  - dimensional em  $\mathbb{R}^n$ . Os vizinhos mais próximos de uma instância são definidos em termos de uma distância padrão, geralmente a distância *Euclidiana*. Mais precisamente, seja uma instância arbitrária descrita por um vetor  $\langle a_1(x), a_2(x), \dots, a_n(x) \rangle$ , no qual  $a_r(x)$  denota o valor do  $r$ -ésimo atributo da instância  $x$ . Então, a diferença entre duas instâncias  $x_i$  e  $x_j$  é definida como  $d(x_i, x_j)$  e é dada pela Equação 5.2 (Mitchell, 1997b):

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (5.2)$$

na qual  $n$  é o número de atributos presentes nas instâncias do conjunto de dados,  $a_r(x_i)$  é o valor do  $r$ -ésimo atributo da instância  $x_i$  e  $a_r(x_j)$  é o valor

do  $r$ -ésimo atributo da instância  $x_j$ . No algoritmo *K-Vizinhos mais Próximos*, a função objetivo (que produz os valores do atributo classe) pode assumir tanto valores discretos como valores contínuos (Mitchell, 1997b). Para problemas com função objetivo de valores discretos dados por uma função  $f : \mathbb{R}^n \rightarrow V$ , na qual  $V$  é o conjunto finito  $\{\nu_1, \nu_2, \dots, \nu_s\}$  de valores que o atributo classe pode assumir, a versão básica do algoritmo *K-Vizinhos mais Próximos* é ilustrada pelo Algoritmo 1. Nesse Algoritmo, o valor  $h(I_c)$  retornado é o valor mais freqüente encontrado entre os  $k$  vizinhos mais próximos da instância a ser classificada ( $I_c$ ). O algoritmo *K-Vizinhos mais Próximos* pode facilmente ser adaptado para problemas com função objetivo de valores contínuos (Mitchell, 1997b).

---

**Algoritmo 1** Versão básica do algoritmo *K-Vizinhos mais Próximos* para problemas com função objetivo de valores discretos

---

**Requer:** Um conjunto de instâncias de treinamento rotuladas  $I = \{I_1, I_2, \dots, I_n\}$ ;

Uma instância de consulta a ser classificada ( $I_c$ );

O número de vizinhos mais próximos a ser considerado ( $k$ );

A função de distância entre instâncias ( $d$ );

**função** *k-VizinhosMaisProximos* ( $I, I_c, k, d$ )

Dado o conjunto de instâncias de treinamento  $I$  armazenado na memória do sistema

Dada a instância  $I_c$  a ser classificada

Determine o conjunto  $J = \{\hat{I}_1, \hat{I}_2, \dots, \hat{I}_k\}$  de  $k$  instâncias mais próximas à instância de consulta  $I_c$ , segundo a função de distância  $d$

$$\text{Faca: } h(I_c) = \underset{v \in V}{\text{argmax}} \sum_{i=1}^k \delta(v, f(\hat{I}_i))$$

na qual  $\delta(a, b) = 1$  se  $a = b$  e  $\delta(a, b) = 0$  caso contrário

**retorne**  $h(I_c)$

**fim função**

---

#### 5.4.7 Algoritmo Kd-Tree

O algoritmo Kd-tree é uma generalização multi-dimensional das árvores de busca binária de uma-dimensão, que facilita o armazenamento e a busca sobre conjuntos de dados de  $k$  dimensões. Essa técnica permite a integração de RBC com Algoritmos de Aprendizado Indutivos e possibilitam uma recuperação por similaridade baseada em Árvores de Decisão (Bentley, 1975).

Uma Kd-Tree define recursivamente o particionamento do espaço dos dados em subespaços disjuntos. Cada nó da árvore representa um subespaço do espaço original dos dados, englobando as instâncias (partículas) pertencentes a esse subespaço. Cada nó não terminal possui dois sucessores, cada um associado a um dos subespaços obtidos pelo particionamento do espaço do por um hiperplano. A abordagem utilizada para o particionamento é a determinação da dimensão para a qual a separação espacial entre os dados é máxima. O espaço do problema é então dividido nesta dimensão em duas regiões (subespaços), cada uma contendo aproximadamente o mesmo número de instâncias. Os subespaços são divididos recursivamente até que os subespaços restantes

tenham somente uma instância. A cada estágio do processo recursivo, o centro de massa da distribuição das instâncias em cada subespaço é calculado (Bentley, 1975). A arquitetura de uma Kd-Tree e o efeito do particionamento do espaço dos dados é ilustrado na Figura 5.6

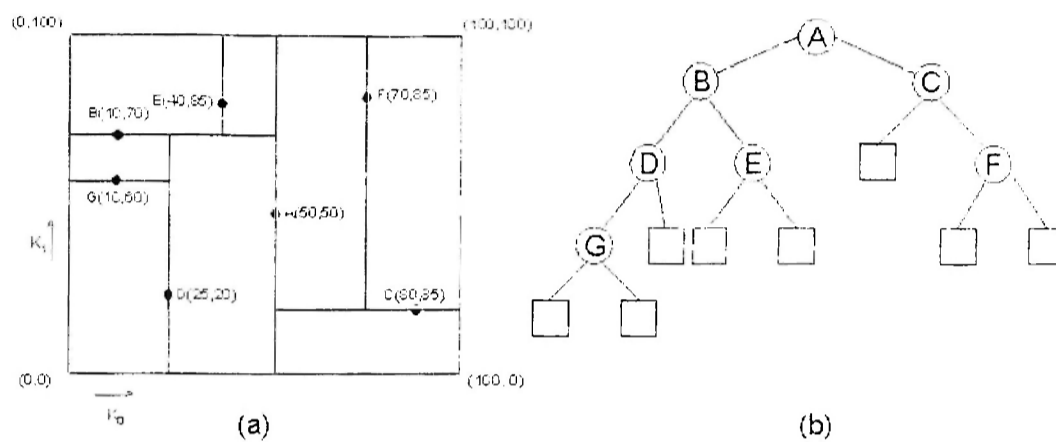


Figura 5.6: Particionamento do espaço dos dados (a) e estrutura da Kd-Tree resultante (b) (Bentley, 1975). Os pontos mostrados em (a) identificam as instâncias armazenadas como nós internos na Kd-Tree e utilizadas para particionar o espaço de busca. As caixas em (a) representam o alcance do subespaço correspondentes às sub-árvores.

## 5.5 Análise de Resultados

Os resultados dos experimentos serão analisados com base na medida do erro absoluto, uma medida muito utilizada na área de AM, que é dada pela Equação 5.3:

$$err(h) = \frac{\sum_{i=1}^n |l_i - o_i|}{n} \quad (5.3)$$

na qual  $l_i$  é a saída desejada para o padrão  $x_i$ ,  $o_i$  é a saída do algoritmo de aprendizado para o  $i$ -ésimo padrão e  $n$  é o número de padrões utilizados para testar o algoritmo de aprendizado.

Visando a estimativa da taxa de erro verdadeira<sup>2</sup> dos algoritmos a serem utilizados, os resultados dos experimentos serão obtidos por meio do método de testes *k-fold-cross-validation*. Nesse método, o conjunto de dados original é dividido aleatoriamente em  $k$  partições mutuamente exclusivas, de tamanho aproximadamente igual. Os padrões de  $k-1$  partições são utilizados no treinamento de um algoritmo de AM, o qual é então testado sobre a partição

<sup>2</sup>A taxa de erro verdadeira de um algoritmo de AM é a taxa calculada sobre o conjunto de exemplos de teste para todo o domínio.

restante. Esse processo é repetido  $k$  vezes, utilizando em cada ciclo uma partição de teste diferente. Para cada partição do teste o erro absoluto do algoritmo é medido. Ao final dos testes com as  $k$  partições, a média do erro absoluto e o desvio padrão do algoritmo são calculados. Supondo que o erro do algoritmo sobre uma partição  $i$  seja  $err(h_i)$ . A média do erro, a variância e o desvio padrão do algoritmo são dados respectivamente pelas Equações 5.4, 5.5 e 5.6 (Monard e Baranauskas, 2003):

$$meam(A) = \frac{1}{k} \sum_{i=1}^k err(h_i) \quad (5.4)$$

$$var(A) = \frac{1}{k} \left[ \frac{1}{k-1} \sum_{i=1}^k (err(h_i) - meam(A))^2 \right] \quad (5.5)$$

$$sd(A) = \sqrt{var(A)} \quad (5.6)$$

Para a comparação entre dois algoritmos  $A$  e  $B$ , foi empregado um teste estatístico para comprovar o desempenho dos resultados dos algoritmos, denominado *Teste t* pareado para procedimentos bilaterais (Moses, 1986; Mitchell, 1997a). Esse teste estatístico permite comprovar com um elevado grau de confiança se o desempenho de um algoritmo é realmente superior ao desempenho de outro. As seguintes etapas devem ser seguidas para a realização desse teste (Moses, 1986; Mitchell, 1997a):

1. particione o conjunto de dados original  $D_0$  em  $k$  sub-conjuntos disjuntos  $T_1, T_2, \dots, T_k$ ;
2. para cada  $i$  de 1 até  $k$ : use  $T_i$  para o conjunto de teste e o restante do conjunto de dados como conjunto de treinamento
  - $S_i \Leftarrow D_0 - T_i$
  - $h_A \Leftarrow L_A(S_i)$
  - $h_B \Leftarrow L_B(S_i)$
  - $d_i \Leftarrow erro_{t_i}(h_A) - erro_{t_i}(h_B)$
3. calcule  $meam(d)$  dado pela Equação 5.7:

$$mean(d) = \frac{1}{k} \sum_{i=1}^k d \quad (5.7)$$

4. calcule  $t_{calc}$  utilizando a Equações 5.8, 5.9 e 5.10:

$$t_{calc} = \frac{meam(d) - 0}{s.e.(meam(d))} \quad (5.8)$$

$$s.e.(mean(d)) = \frac{s_d}{\sqrt{k}} \quad (5.9)$$

$$s_d = \sqrt{\frac{\sum_{i=1}^k (d_i - mean(d))^2}{k-1}} \quad (5.10)$$

5. determinar os limites da região de aceitação  $RA : (-k_{\frac{\alpha}{2}, k-1}, k_{\frac{\alpha}{2}, k-1})$  na tabela de distribuição de *t Student* com base no valor do nível de significância ( $\alpha$ ) e no valor de  $k - 1$ ;

Se o valor de  $t_{calc}$  estiver dentro da  $RA$ , então os algoritmos possuem desempenho equivalentes. Se o valor de  $t_{calc}$  for positivo e estiver fora da  $RA$ , então o algoritmo  $B$  é melhor que o algoritmo  $A$  com  $X\%$  de confiança. Se o valor de  $t_{calc}$  for negativo e estiver fora da  $RA$ , então o algoritmo  $A$  é melhor que o algoritmo  $B$  com  $X\%$  de confiança.

## 5.6 Considerações Finais

Neste Capítulo foram apresentados alguns conceitos sobre AM, que foram estudados durante o desenvolvimento dos experimentos deste trabalho de Mestrado.

Foram discutidas brevemente as técnicas utilizadas para o pré-processamento de alguns conjuntos de dados utilizados nos experimentos realizados.

Também foram apresentados os algoritmos utilizados durante os experimentos realizados nesse trabalho de Mestrado.

Finalmente, foram descritos os métodos utilizados para a avaliação dos resultados obtidos nos experimentos realizados.



---

## Abordagem Híbrida de Adaptação

---

### 6.1 Considerações Iniciais

A adaptação de casos é um dos maiores desafios da área de RBC. Em contraste com a aquisição de casos, a aquisição de conhecimento necessário para a adaptação de casos exige um alto esforço de aquisição (Hanney, 1996; Wiratunga et al., 2002).

A forma de adaptação mais utilizada é a codificação de regras de adaptação, demandando um significativo esforço para a aquisição de conhecimento (Hanney, 1996). Por exemplo, um sistema de RBC proposto em (Leake et al., 1996) (Subseção 3.2 do Capítulo 3) possui um conjunto de regras de adaptação e um método para busca em memória. Quando um novo problema é apresentado ao sistema, ele recupera um caso similar o novo problema e o envia ao mecanismo de adaptação, que seleciona uma regra de adaptação apropriada e ativa o mecanismo de busca em memória. Essas regras de adaptação são pacotes de conhecimento codificados e específicos para um domínio da aplicação.

Uma alternativa para superar as dificuldades em adquirir conhecimento para adaptação de casos tem sido a utilização de aprendizado automático. Neste Capítulo são discutidos os experimentos realizados com uma abordagem híbrida de adaptação de casos capaz de efetuar a aquisição automática de conhecimento e utilizar esse conhecimento para a adaptação automática das soluções recuperadas.

Na Seção 6.2 é apresentada a abordagem híbrida de adaptação de casos investigada neste trabalho. Na Seção 6.3 é mostrada a evolução da abordagem investigada com a utilização de comitês de algoritmos de AM para a adaptação de casos. Na Seção 6.4 é ilustrado o funcionamento do mecanismo híbrido

de recuperação de casos construído para os experimentos realizados com a abordagem de adaptação de casos. Na Seção 6.5 são descritos os conjuntos de dados utilizados nos experimentos realizados. Na Seção 6.6 são ilustrados os passos dos experimentos conduzidos. Na Seção 6.7 são apresentados os resultados obtidos pelos experimentos realizados. Na Seção 6.8, os resultados obtidos nos experimentos são analisados. Na Seção 6.9 são feitas as considerações finais do capítulo.

## 6.2 Abordagem Híbrida de Adaptação de Casos

Este trabalho investigou o uso de algoritmos de AM para realizar a adaptação automática das soluções recuperadas de uma BC. Foram investigados algoritmos de AM (nomeados *estimadores*), baseados em diferentes paradigmas, para compor individualmente o mecanismo de adaptação de casos. Os estimadores foram utilizados para realizar adaptações em domínios com componentes<sup>1</sup> de soluções numéricos ou simbólicos. Diferentes versões da abordagem de adaptação de casos foram investigadas, empregando diferentes algoritmos de AM para compor o mecanismo de adaptação de casos:

- a Rede Neural Artificial do tipo MLP (Haykin, 1999);
- os algoritmos de aprendizado simbólico C4.5 (Quinlan, 1993) e M5 (Quinlan, 1992; Wang e Witten, 1997);
- a técnica SVM (Vapnik, 1998), baseada na teoria de aprendizado estatístico.

Os algoritmos de aprendizado simbólico (C4.5 e M5) foram utilizados alternadamente, dependendo do componente da solução sendo adaptado. Para os componentes contínuos da solução, foi utilizado o algoritmo M5. Para os componentes discretos da solução, foi utilizado o algoritmo C4.5.

A abordagem investigada para adaptação de casos emprega dois módulos:

1. gerador de padrões de adaptação – produz um conjunto de dados formado por padrões de adaptação. Esse conjunto de dados é, em seguida, utilizado pelo segundo módulo.
2. mecanismo de adaptação de casos – realiza automaticamente a adaptação das soluções recuperadas, utilizando o conhecimento produzido pelo primeiro módulo.

---

<sup>1</sup>Neste trabalho, os atributos das soluções são nomeados componentes.



Essa abordagem assume que a BC original é representativa (Smyth, 1998), ou seja, a BC é uma boa representação dos problemas do domínio da aplicação. Essa característica deve ser garantida durante a aquisição dos casos em um processo de análise de competência da BC. Portanto, nenhum treinamento adicional é requerido quando novos casos são adicionados à BC durante o processo de raciocínio. Adicionalmente, pode-se utilizar uma política de manutenção de conhecimento (Smyth e Keane, 1995; Smyth, 1998; Zhu e Yang, 1999; Leake e Wilson, 2000) para monitorar a evolução da BC do sistema e acionar o processo que efetua novamente o treinamento dos algoritmos de AM quando necessário.

### 6.2.1 Geração dos Padrões de Adaptação

O módulo gerador de padrões de adaptação é capaz de extrair conhecimento implícito de uma BC. O processo de geração dos padrões de adaptação é mostrado no Algoritmo 2.

---

#### **Algoritmo 2** Geração de Padrões de Adaptação.

---

**Requer:** o número de casos similares a serem recuperados (*NumeroCasos*);  
 O componente da solução que será utilizado para a construção dos padrões de adaptação (*Componente*).

**função** GeraPadraoAdaptacao (*NumeroCasos*, *Componente*)

**para cada** caso da base de casos original **faça**

*CasoProva* ← *ExtraiCasoProva* ()

*DescProva* ← *ExtraiDescricao* (*CasoProva*)

*SolucaoProva* ← *ExtraiSolucao* (*CasoProva*, *Componente*)

*CasosRecuperados* ← *Recupera* (*DescProva*, *NumeroCasos*)

**para cada** caso em *CasosRecuperados* **faça**

*DescRecuperada* ← *ExtraiDescricao* (*CasosRecuperados*(*i*))

*SolucaoRecuperada* ← *ExtraiSolucao* (*CasosRecuperados*(*i*), *Componente*)

*ConstroiPadraoAdaptacao* (*DescProva*, *DescRecuperada*, *SolucaoRecuperada*, *SolucaoProva*)

**fim para**

**fim para**

**fim função**

---

Inicialmente, o algoritmo de geração de padrões extrai um caso da BC original e o utiliza como um novo problema (*CasoProva*) apresentado ao sistema. Os demais casos formam uma nova BC, sem o caso prova. Em seguida, o algoritmo extrai os atributos da descrição do problema contido no caso prova (*DescProva*) e um componente (indicado por *Componente*) da solução do caso prova (*SolucaoProva*). Então, esse algoritmo retorna os *NumeroCasos* casos mais similares à descrição do problema do caso prova (*CasosRecuperados*). Para cada caso recuperado, os atributos da descrição do problema (*DescRecuperada*) e o componente correspondente da solução (indicado por *Componente*) são extraídos (*SolucaoRecuperada*). Em seguida, o algoritmo gera um padrão de adaptação utilizando as seguintes partes do caso prova e do caso recuperado:

- atributos de entrada:
  - a descrição do problema armazenado no caso prova;

- a descrição do problema armazenado no caso recuperado;
- o componente da solução (indicado por *Componente*) do caso recuperado.
- atributo de saída:
  - o componente da solução (indicado por *Componente*) do caso prova.

Finalmente, o conjunto de padrões de adaptação gerado é utilizado para o treinamento dos algoritmos de AM (MLP, SVM e C4.5 ou M5). O processo de treinamento é parte integrante da abordagem de adaptação.

### 6.2.2 Mecanismo de Adaptação de Casos

O mecanismo de adaptação de casos investigado possibilita o aprendizado das modificações que precisam ser realizadas nos valores dos componentes de uma solução recuperada de maneira a obter uma solução adequada para um novo problema. A característica mais importante desse mecanismo é o emprego do conhecimento obtido a partir de uma BC com um esforço reduzido para a aquisição do conhecimento. O processo de adaptação de casos é mostrado no Algoritmo 3.

---

#### **Algoritmo 3** Mecanismo de Adaptação de Casos.

---

**Requer:** a descrição do novo problema apresentado ao sistema (*Descricao*);  
 O caso recuperado pelo mecanismo de recuperação (*CasoRecuperado*);  
 O componente da solução que será adaptado (*Componente*).

**função** Adaptacao (*Descricao*, *CasoRecuperado*, *Componente*)  
*DescRecuperada*  $\Leftarrow$  ExtraiDescricao (*CasoRecuperado*)  
*SolucaoRecuperada*  $\Leftarrow$  ExtraiSolucao (*CasoRecuperado*, *Componente*)  
*PadraoEntrada*  $\Leftarrow$  ConstroiPadraoEntrada (*Descricao*, *DescRecuperada*, *SolucaoRecuperada*)  
*Acao*  $\Leftarrow$  MecanismoAdaptacao (*Normalizacao*(*PadraoEntrada*), *Componente*)  
*NovaSolucao*  $\Leftarrow$  AplicaAcao (*SolucaoRecuperada*, *Acao*, *Componente*)  
**retorne** *NovaSolucao*  
**fim função**

---

Quando um novo problema é apresentado ao sistema de RBC, o caso mais similar é recuperado, por um mecanismo de recuperação (Lenz, 1996a; Duda e Stork, 2001). Esse caso recuperado (*CasoRecuperado*) é enviado ao mecanismo de adaptação juntamente com a descrição do novo problema (*Descricao*). O algoritmo de adaptação, por sua vez, extrai os atributos da descrição do problema contido no caso recuperado (*DescRecuperada*). Em seguida, o algoritmo extrai um componente (indicado por *Componente*) da solução contida no caso recuperado (*SolucaoRecuperada*). Então, o algoritmo gera um padrão de entrada no formato adequado e apresenta esse padrão para o estimador desenvolvido para o componente atual (*MecanismoAdaptacao*). Em seguida, o estimador indica a modificação necessária ao componente atual (*Acao*). Finalmente, a modificação indicada é aplicada ao componente da solução recuperada para gerar um novo componente para a solução do novo problema (*NovaSolucao*).

Para a aplicação dessa abordagem híbrida de adaptação, foi adotada uma estratégia na qual deve ser gerado um conjunto de padrões de adaptação independente e deve ser utilizado um mecanismo de adaptação distinto para cada componente da estrutura da solução dos casos em um domínio. Essa estratégia mantém a independência da abordagem de adaptação em relação à estrutura da solução dos casos.

A Figura 6.1 ilustra um exemplo da geração de um padrão de adaptação para um dos conjuntos de dados utilizados durante os experimentos realizados, o conjunto de dados *Bridges*. A descrição desse conjunto de dados pode ser encontrada na Seção 6.5. Para esse domínio, as soluções dos casos são compostas por cinco componentes. Seguindo a estratégia de gerar um conjunto de padrões e adotar um mecanismo de adaptação distinto para cada componente da solução, para esse conjunto de dados foram gerados cinco conjuntos independentes de padrões de adaptação e foram utilizados cinco mecanismos de adaptação distintos, um para cada componente da solução. Esse exemplo ilustra um padrão de adaptação gerado para o componente *material* da solução dos casos:

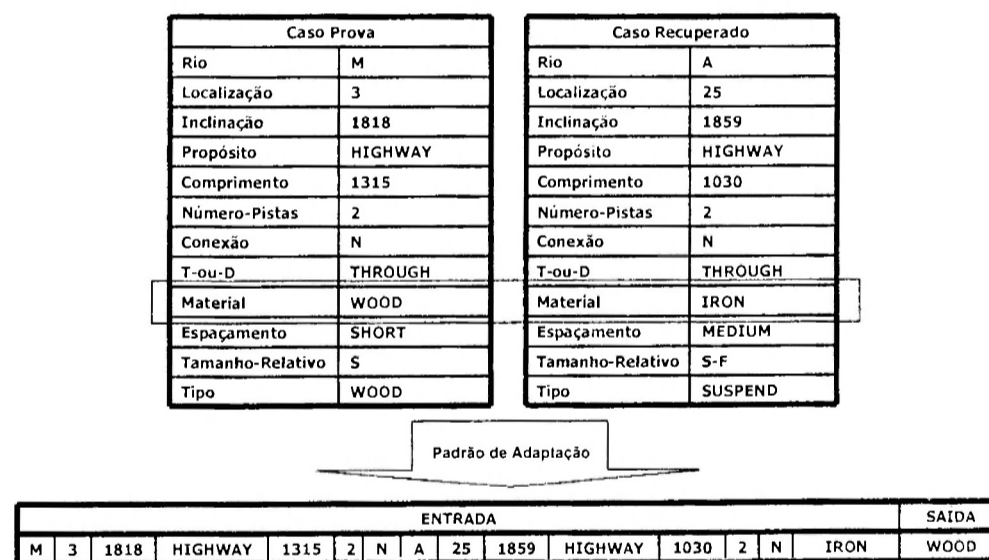


Figura 6.1: Exemplo do processo de geração dos padrões de adaptação.

Os conjuntos de padrões de adaptação gerados são então utilizados para treinar os algoritmos de AM que são então empregados como mecanismos de adaptação de casos.

### 6.3 Comitês de Algoritmos de AM

Na Seção 6.2 foi apresentada a abordagem híbrida de adaptação de casos, investigada inicialmente neste trabalho de mestrado. Tal abordagem utiliza

algoritmos de AM individualmente (estimadores) para compor o mecanismo de adaptação. Após a obtenção dos primeiros resultados pelos experimentos com a abordagem híbrida, foi investigada uma evolução do mecanismo de adaptação, pela substituição dos algoritmos individuais de AM por comitês de algoritmos de AM. O objetivo dessa evolução foi a busca por melhor acurácia e precisão

Os comitês investigados são compostos por algoritmos de AM (estimadores), baseados em diferentes paradigmas. Adicionalmente, um algoritmo de AM, denominado *combinador*, combina as saídas dos estimadores para produzir a saída do comitê. Os estimadores e o combinador são utilizados para realizar adaptações em domínios com componentes de soluções numéricos ou simbólicos. Os Comitês são compostos pelos seguintes algoritmos de AM:

- estimadores: A RNA do tipo MLP (Haykin, 1999); os algoritmos de aprendizado simbólico C4.5 (Quinlan, 1993) e M5 (Quinlan, 1992; Wang e Witten, 1997); a técnica SVM (Vapnik, 1998).
- combinador: neste trabalho foram investigados os mesmos quatro algoritmos de AM utilizados como estimadores para exercer o papel de combinador do comitê – A RNA do tipo MLP, A técnica SVM e os algoritmos de aprendizado simbólico C4.5 e M5. O combinador recebe como entrada as saídas dos estimadores e produz a saída do comitê.

Os algoritmos de aprendizado simbólico (C4.5 e M5) foram utilizados alternadamente, dependendo do componente da solução sendo adaptado. Para os componentes contínuos da solução, foi utilizado o algoritmo M5. Para os componentes discretos da solução, foi utilizado o algoritmo C4.5. Portanto, em qualquer caso, o comitê é composto sempre por três estimadores (uma rede MLP, um SVM e um C4.5 ou um M5) e por um combinador (uma rede MLP ou um SVM ou um C4.5 ou um M5). A Figura 6.2 ilustra a arquitetura dos comitês de algoritmos de AM investigados.

Os comitês foram empregados como mecanismos de adaptação, substituindo os algoritmos de AM utilizados individualmente, constituindo novas versões da abordagem de adaptação de casos. Essas novas versões foram comparadas com as versões empregando algoritmos de AM individualmente, com o objetivo de verificar a hipótese de que os comitês melhoram o desempenho da abordagem híbrida de adaptação casos, aumentando a acurácia (reduzindo média do erro) e a precisão (reduzindo desvio padrão dos erros) do sistema de RBC.

Para essas novas versões da abordagem híbridas, o Algoritmo 2 foi mantido sem alterações, o Algoritmo 3 foi alterado para incorporar os comitês como

#### 6.4 Abordagem Híbrida de Recuperação de Casos

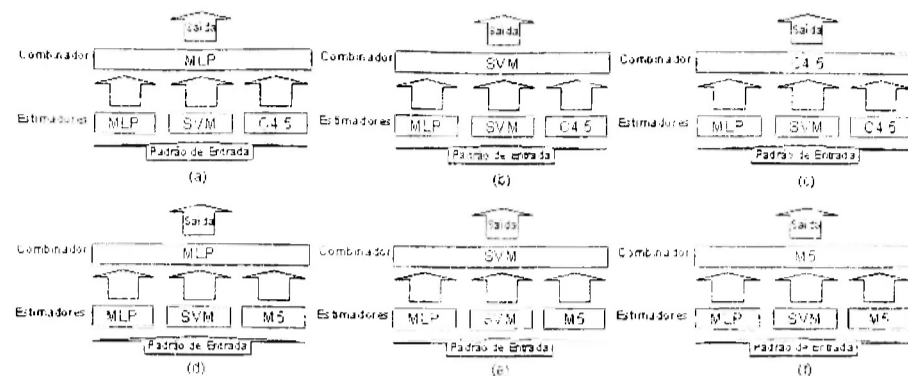


Figura 6.2: Arquitetura dos comitês investigados. Os comitês (a), (b) e (c) foram utilizados para a adaptação de componentes discretos. Os comitês (d), (e) e (f) foram utilizados para a adaptação de componentes contínuos.

mecanismo de adaptação (ver *Mecanismo Adaptação* no Algoritmo 3). O treinamento do mecanismo de adaptação com o conjunto de padrões de adaptação também foi modificado. Primeiramente, os estimadores (MLP, SVM e C4.5 ou M5) são treinados individualmente. Em seguida, as saídas desses estimadores são utilizadas para treinar o combinador do comitê.

#### 6.4 Abordagem Híbrida de Recuperação de Casos

Para a realização dos experimentos com as versões da abordagem de adaptação investigadas nesta Dissertação de Mestrado, foram utilizadas duas abordagens diferentes de recuperação de casos. Tal escolha teve por objetivo investigar a influência do recuperador no processo de adaptação e verificar a capacidade de integração da abordagem híbrida de adaptação de casos com diferentes recuperadores. Para tanto, foi selecionada a abordagem clássica de recuperação de casos *Kd-Tree* (Bentley, 1975), discutido na Seção 5.4 do Capítulo 5. Além disso, foi desenvolvida uma abordagem híbrida de recuperação que utiliza uma RNA do tipo ART2, apresentada nesta Seção.

A abordagem híbrida de indexação e recuperação de casos contém dois níveis de organização de memória, como ilustrado na Figura 6.3: o primeiro nível de memória é formado pela camada de saída de uma rede ART2 e é utilizado para criar e indicar o agrupamento de casos mais similares ao novo problema apresentado, reduzindo o espaço de busca e o tempo de recuperação. O segundo nível de memória é formado por uma simples memória plana que armazena as instâncias dos casos organizados pelos agrupamentos criados pelo primeiro nível.

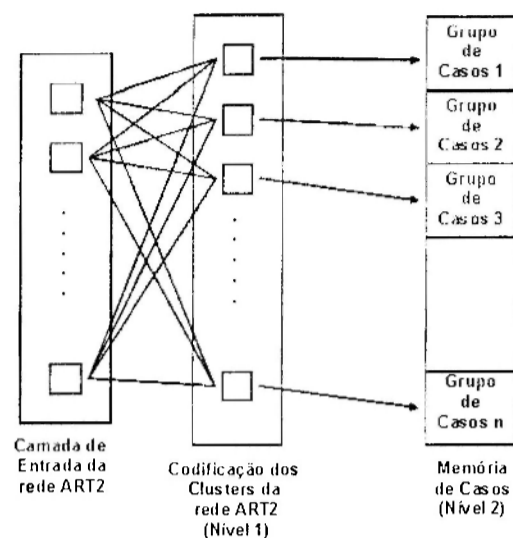


Figura 6.3: Arquitetura do Mecanismo de Indexação e Recuperação de Casos.

#### 6.4.1 Processo de Indexação

O mecanismo de indexação de casos permite o armazenamento de novos casos a qualquer momento que o sistema necessite aprender. A organização de memória desse mecanismo possibilita o armazenamento de novos casos sem a eliminação dos casos já armazenados. O processo de indexação é mostrado no Algoritmo 4.

---

#### Algoritmo 4 Indexação de Novos Casos.

---

**Requer:** o novo caso a ser armazenado na memória do sistema (*NovoCaso*).

**função** *ArmazenaNovoCaso(NovoCaso)*  
*Descricao* ← *ExtraiDescricao(NovoCaso)*  
*CodigoGrupo* ← *TreinaRedeART2(PreProcessa(Descricao))*  
*SalvaNovoCaso(NovoCaso, CodigoGrupo)*

**fim função**

---

Quando o sistema precisa aprender um novo caso (*NovoCaso*), esse caso é enviado para o mecanismo de indexação, que extrai a descrição do problema (*Descricao*) e a salva na forma de um vetor de pares atributo-valor. Depois, esse vetor é pré-processado para formar o padrão de entrada utilizado para treinar a rede ART2. Após o treinamento, a rede ART2 indica o agrupamento ao qual o novo caso pertence (*CodigoGrupo*). O mecanismo de indexação armazena o novo caso na memória plana (nível 2) no agrupamento de similaridade indicado pela rede ART2.

#### 6.4.2 Processo de Recuperação

O mecanismo de recuperação de casos permite a recuperação de um ou mais casos prévios similares, de acordo com as necessidades do sistema. A organização de memória desse mecanismo, utilizando a rede ART2, permite a

## 6.5 Descrição dos Conjuntos de Dados

---

divisão do espaço de busca do algoritmo, reduzindo o tempo de recuperação. O processo de recuperação é ilustrado no Algoritmo 5.

---

### Algoritmo 5 Recuperação de Casos.

---

**Requer:** a descrição do novo problema apresentado ao sistema (*Descricao*);  
O número de casos similares a serem recuperados (*NumeroCasos*).

**função** *Recupera*(*Descricao*, *NumeroCasos*)  
*CodigoGrupo* ← *TestaRede*.ART2(*PreProcessa*(*Descricao*))  
*CasosSelecioneados* := *RecuperaGrupo*(*CodigoGrupo*)  
*CasosSimilares* := *CalculaSimilaridade*(*Selecioneados*, *Descricao*)  
*CasosRecuperados* := *ConstroiRanking*(*Similares*, *NumeroCasos*)  
**retorne** *Recuperados*  
**fim função**

---

Quando o sistema precisa recuperar casos similares a um novo problema apresentado, ele extrai a descrição do novo problema e envia para o mecanismo de recuperação na forma de um vetor de pares atributo-valor (*Descricao*). O mecanismo de recuperação, por sua vez, pré-processa esse vetor para formar um padrão de entrada que é apresentado à rede ART2, que indica o agrupamento (*CodigoGrupo*) mais similar a esse padrão. O mecanismo de recuperação seleciona os casos armazenados pertencentes ao agrupamento indicado pela rede ART2 (*CasosSelecioneados*). A similaridade entre cada caso do agrupamento selecionado e o novo problema (*CasosSimilares*) é calculada utilizando o algoritmo *K - Vizinhos mais Próximos*. O mecanismo de recuperação constrói um ranking dos casos mais similares (*CasosRecuperados*) e retorna o conjunto de casos mais similares. O número de casos mais similares a serem recuperados é indicado por *NumeroCasos*.

## 6.5 Descrição dos Conjuntos de Dados

Nesta Seção são descritos os seis conjuntos de dados utilizados para realizar as avaliações empíricas das versões da abordagem de adaptação de casos investigadas. Os conjuntos de dados *Servo*, *Imports*, *Housing* e *Bridges* podem ser obtidos no repositório de dados UCI (Blake e Merz, 1998). O conjunto de dados sobre qualidade de água foi obtido na página de descoberta de conhecimento em bases de dados, também do repositório UCI. O conjunto de dados *Vinho* foi obtido junto à Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) Instrumentação de São Carlos. Para cada conjunto de dados será feita uma breve descrição do seu propósito e apresentada uma tabela que descreve seus atributos.

**Servo** Esse conjunto de dados foi extraído a partir da simulação de um sistema de servo-mecanismo envolvendo um amplificador servo, um motor e um carrinho deslizante. O valor da saída é o tempo requerido para o sistema responder aos passos em um determinado conjunto de posições.

Esse conjunto de dados foi utilizado por outros autores para a avaliação de outras abordagens para adaptação do casos (Hanney, 1996). A Tabela 6.1 ilustra a estrutura dos casos desse conjunto de dados.

Tabela 6.1: Estrutura dos Casos de Servo Mecanismo.

	Atributo	Valor
Problema	Motor	A,B,C,D,E
	Parafuso	A,B,C,D,E
	GanhoP	3,4,5,6
	GanhoV	1,2,3,4,5
Solução	Classe	Continue

**Imports** Esse conjunto de dados é utilizado para a predição do valor de um veículo baseada na especificação de várias características do veículo. A Tabela 6.2 na próxima página ilustra a estrutura dos casos desse conjunto de dados.

**Housing** Esse conjunto de dados é utilizado para a predição do valor médio de moradia nos subúrbios de Boston dado um conjunto de características da moradia. A Tabela 6.3 na página 72 ilustra a estrutura dos casos desse conjunto de dados.

**Bridges** Esse conjunto de dados está relacionado com o projeto e especificação pontes da cidade de Pittsburgh. Nesse conjunto de dados não existem classes a serem preditas, mas sim 5 propriedades (descrição do projeto) que precisam ser preditas com base em 7 especificações. A Tabela 6.4 na página 72 ilustra a estrutura dos casos desse conjunto de dados.

**Water** Esse conjunto de dados foi empregado na Competição de Inteligência Computacional e Aprendizado de 1999. Os dados surgiram a partir de um estudo de qualidade de água em que as amostra foram coletadas em diferentes pontos de vários rios da Europa por um período de aproximadamente um ano. Foram analisadas várias substâncias químicas relacionadas com a distribuição de diferentes populações de algas. A Tabela 6.5 na página 73 ilustra a estrutura dos casos desse conjunto de dados.  $CC_s$  indica a concentração da s-ésima substância química.  $AG_m$  indica a distribuição da m-ésima população de alga.

**Vinho** Esse conjunto de dados está relacionado com o problema da predição de características de vinhos com base da medição de 9 sensores químicos. Esses dados foram obtidos durante os testes do projeto da Língua Eletrônica da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA)



Tabela 6.2: Estrutura dos Casos de Carros Importados.

	Atributo	Valor
Problema	Risco	-3, -2, -1, 0, 1, 2, 3
	Perda-Normalizada	Continue
	Fabricante	alfa-romero, audi, bmw, chevrolet, dodge, ...
	Tipo-Combustivel	diesel, gás
	Aspiração	std, turbo
	Número-Portas	four, two
	Estilo-Carroceria	hardtop, wagon, sedan, hatchback, convertible
	Tração	4wd, fwd, rwd
	Localização-Motor	front, rear
	Relação-Rodagem	Continue
	Largura	Continue
	Comprimento	Continue
	Altura	Continue
	Limite-Peso	Continue
	Tipo-Motor	dohc, dohcv, l, ohc, ohcv, rotor
	Número-Cilindros	eight, five, four, six, three, twelve, two
	Tamanho-Motor	Continue
	Sistema-Combustivel	1bbl, 2bbl, 4bbl, idi, mfi, mpti, spdi, spli
	Cilindrada	Continue
	Percurso	Continue
Taxa-Compressão	Continue	
Potência	Continue	
Rotação-Máxima	Continue	
Consumo-Cidade	Continue	
Consumo-Estrada	Continue	
Solução	Preço	Continue

de São Carlos. A Tabela 6.6 na página 73 ilustra a estrutura dos casos desse conjunto de dados.

As principais características destes conjuntos de dados são resumidas na Tabela 6.7. A coluna *Exemplos* traz o número de exemplos no conjunto de dados original. A coluna *Atributos(Contínuo, Simbólico)* mostra o número total de atributos e o número de atributos discretos e contínuos. A coluna *Atributos Ausentes* indica o número de atributos ausentes no conjunto original. A coluna *Duplicados ou Conflitantes* indica o número de exemplos duplicados somado do número de exemplos com mesmo conjunto de atributos de entrada mas com atributos de saída diferentes. A coluna *Erro Majoritário* indica o limite de erro para os atributos de saída discretos.

A Figura 6.4 exhibe o gráfico contendo as distribuições dos valores dos componentes das soluções dos conjuntos de dados *Servo*, *Imports*, *Housing*, *Water* (*Algal 1, ..., Algal 7*) e *Vinho* (*Indice, Nota e PH*). De uma maneira geral,

Tabela 6.3: Estrutura dos Casos de Moradia.

	Atributo	Valor
Problema	taxa per capita de crime	Continue
	Proporção de terras residenciais	Continue
	Proporção de acres não negociados	Continue
	Variável de Charles River	Continue
	Concentração de óxido nítrico	Continue
	Número médio dos quartos por a moradia	Continue
	Proporção de unidades ocupadas por proprietários	Continue
	Distância ponderada a cinco centros de Boston	Continue
	Índice de acessibilidade à auto-vias radiais	Continue
	Taxa de impostos sobre propriedade por \$10,000	Continue
	Taxa de ensino	Continue
	Proporção de negros	Continue
	% população de baixa renda	Continue
	Solução	Valor médio da moradia

Tabela 6.4: Estrutura dos Casos de Pontes de Pittsburgh.

	Atributo	Valor
Problema	Rio	a, m, o
	Localização	1..52
	Inclinação	1818..1986
	Propósito	walk, aqueduct, rr, highway
	Comprimento	804..4558
	Número-Pistas	1, 2, 4, 6
	Conexão	N, G
Solução	T-ou-D	through, deck
	Material	wood, iron, steel
	Espaçamento	short, medium, long
	Tamanho-Relativo	S, S-F, F
	Tipo	wood, suspend, simple-t, arch, cantilev, cont-t

o gráfico mostra uma distribuição semelhante para os valores do componentes das soluções dos conjuntos de dados *Servo*, *Imports*, *Housing* e *Water*, com uma maior concentração dos valores de saída no intervalo [0. . . 0.1].

## 6.6 Descrição dos Experimentos

Esta Seção apresenta o conjunto de experimentos realizados para a avaliação empírica do desempenho das versões da abordagem de adaptação automática de casos investigadas. Para isso, o desempenho obtido pelos diferentes sistemas híbridos de RBC (utilizando os diferentes algoritmos de AM indivi-

Tabela 6.5: Estrutura dos Casos de Qualidade de Água.

	Atributo	Valor
Problema	Estação do Ano	summer, spring, autumn, winter
	Tamanho do rio	small, medium, large
	velocidade da água	high, medium, low
	$CC_1$	Continue
	...	...
Solução	$Alg_1$	Continue
	...	...
	$AG_7$	Continue

Tabela 6.6: Estrutura dos Casos da Base de Vinho.

	Atributo	Valor
Problema	Tipo do vinho	white, red
	Sensor 1	Continue
	Sensor 2	Continue
	Sensor 3	Continue
	Sensor 4	Continue
	Sensor 5	Continue
	Sensor 6	Continue
	Sensor 7	Continue
	Sensor 8	Continue
	Sensor 9	Continue
Solução	PH	Continue
	Índice de absorção	Continue
	Nota	61..88

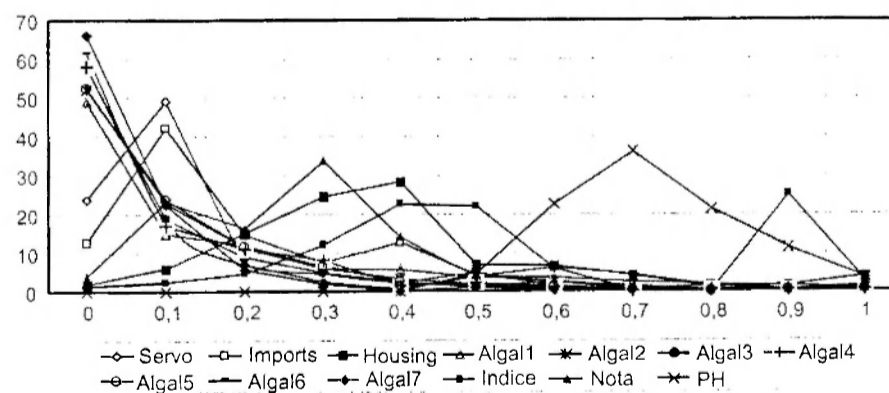


Figura 6.4: Distribuição dos valores das soluções dos conjuntos de dados com valores de saída contínuos.

duais e comitês como mecanismos de adaptação e as duas abordagens de recuperação de casos) foram comparados. Para mostrar que a adaptação au-

Tabela 6.7: Características dos Conjuntos de Dados.

Conjunto de Dados	Exemplos	Atributos (Contínuo, Simbólico)	Atributos Ausentes	Duplicados ou Conflitantes	Erro Majoritário (%)
Servo	167	4(0.4)	0	0	
Imports	205	25(14.11)	46	8	
Housing	506	13(13.0)	0	0	
Bridges	108	12(2.10)	38	0	T-ou-D = 14.61 Material = 21.35 Espaçamento = 43.82 Tamanho-Relativo = 43.82 Tipo = 59.46
Water	340	18(15.3)	58	0	
Vinhos	289	13(12.1)	0	2	

tomática de casos pode resultar em um considerável ganho na predição dos valores desejáveis para os componentes de solução dos casos. o desempenho (em termos de acurácia e precisão) das diferentes versões dos sistemas híbridos de RBC foram comparadas com o desempenho obtido por uma rede MLP, o algoritmo SVM, o algoritmo C4.5 e o algoritmo M5 aplicados isoladamente para a predição de valores sobre os mesmos conjuntos de dados.

As RNAs empregadas nos experimentos foram treinadas utilizando o algoritmo momentum backpropagation com termo de momentum igual a 0.2 e taxa de aprendizado igual a 0.3. As topologias<sup>2</sup> das RNAs utilizadas como combinadores são mostradas na Tabela 6.8.

Tabela 6.8: Topologias das Redes Neurais Artificiais Empregadas como Combinadores.

Conjunto de Dados	Entrada	Camada intermediária	Saída
Servo	3	2	1
Imports	3	2	1
Housing	3	2	1
Bridges	3	2	2...5
Water	3	2	1
Vinhos	3	2	1

As topologias das RNAs utilizadas como estimadores são mostradas na Tabela 6.9.

O algoritmo M5 foi utilizado para geração de classificadores do tipo *Model Tree* com fator de poda igual a 2. O algoritmo C4.5 foi utilizado com os parâ-

<sup>2</sup>As topologias mais promissoras para as RNAs foram obtidas automaticamente pela ferramenta WEKA, por meio da análise do conjunto de dados, com a utilização da opção *autobuild*.

Tabela 6.9: Topologias das Redes Neurais Artificiais Empregadas como Estimadores.

Conjunto de Dados	Entrada	Camada intermediária	Saída
Servo	25	13	1
Imports	162	82	1
Housing	27	14	1
Bridges	29...33	30	2...5
Water	37	20	1
Vinhos	23	12	1

metros padrão. O algoritmo SVM foi utilizado com kernel RBF e parâmetros padrão.

A rede MLP, o algoritmo M5 e o algoritmo C4.5 foram simulados utilizando a ferramenta WEKA<sup>3</sup>. O algoritmo SVM foi simulado utilizando a ferramenta LIBSVM<sup>4</sup>.

Três diferentes conjuntos de padrões de adaptação foram utilizados durante os experimentos. Esses três conjuntos foram construídos pela variação do número de casos similares ao caso prova considerados durante a geração dos padrões de adaptação (variando o parâmetro *NumeroCasos* no Algoritmo 2). Os conjuntos de padrões gerados consideram respectivamente um, três e cinco casos similares.

O pré-processamento dos conjuntos de dados incluiu a normalização dos valores numéricos de entrada (ver *Normalizacao* no Algoritmo 3) para o intervalo  $[0 \dots 1]$ . Os valores simbólicos de entrada foram transformados em vetores ortogonais de valores binários. Para a rede MLP e o algoritmo SVM, os componentes das soluções com valores simbólicos foram transformados da mesma maneira. Adicionalmente, para o conjunto de dados de projetos de pontes de Pittsburgh, os atributos de entrada ausentes foram preenchidos utilizando a técnica proposta em (Batista, 2003). Esse procedimento foi adotado para impedir a eliminação dos exemplos contendo atributos ausentes, uma vez que esse conjunto de dados possui poucos exemplos. Após o preenchimento dos valores de entrada ausentes, o conjunto de dados resultante foi balanceado utilizando a técnica de *over-sampling* proposta em (Chawla et al., 2002). Esse procedimento foi adotado para aprimorar o desempenho do mecanismo de adaptação, uma vez que os algoritmos de AM podem apresentar dificuldades com conjuntos de dados não balanceados.

Durante os experimentos, foram utilizadas duas abordagens diferentes para a recuperação de casos: a abordagem *Kd-Tree* e a abordagem híbrida dis-

<sup>3</sup>Disponível em <http://www.cs.waikato.ac.nz/ml/weka/index.htm>

<sup>4</sup>Disponível em <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

cutida na Seção 6.4, com o objetivo de investigar a influência do recuperador de casos no processo de adaptação e verificar a capacidade de integrar a abordagem híbrida de adaptação de casos com diferentes recuperadores.

Os experimentos seguiram a metodologia *10 – fold – cross – validation*. Os exemplos do conjunto de dados foram divididos randomicamente em 10 grupos (*folds*) com tamanhos aproximadamente iguais. Um *fold* foi utilizado como exemplos de teste - *fold de teste* (um conjunto de novos problemas a serem apresentados ao sistema) e os 9 *folds* restantes foram considerados para treinamento - *fold de treinamento* (um conjunto de casos prévios armazenados na BC do sistema). Cada par (*fold de teste*, *fold de treinamento*) formou uma iteração para a validação dos algoritmos, num total de 10 iterações. Para cada interação, antes do treinamento dos mecanismos de adaptação (algoritmos de AM e comitês) utilizando o *fold de treinamento*, os exemplos desse conjunto de dados foram pré-processados conforme o procedimento de pré-processamento discutido acima. Após o treinamento mecanismo de adaptação utilizando o *fold de treinamento*, o *fold de teste* foi apresentado ao sistema e o erro absoluto médio ou o erro de classificação foi calculado (o erro absoluto médio foi calculado para os componentes numéricos da solução dos casos e o erro de classificação foi calculado para os componentes simbólicos da solução dos casos). Esse processo foi repetido para os 9 *folds* restantes. Em seguida, a média e o desvio padrão dos erros obtidos para cada uma das 10 iterações foi calculado. A Figura 6.5 ilustra a metodologia seguida para a realização dos experimentos.

## 6.7 Resultados Empíricos

Nesta Seção são apresentados os resultados obtidos durante os experimentos realizados para a avaliação empírica das versões da abordagem de adaptação automática de casos investigadas. Os experimentos comparam o desempenho dos sistemas híbridos de RBC – utilizando como mecanismo de adaptação os algoritmos de AM individuais ou os comitês – com o desempenho dos classificadores individuais (MLP, SVM e C4.5 ou M5). As Tabelas de resultados mostram as taxas de erro absoluto médio (para conjuntos de dados com componentes contínuos), ou de erro médio de classificação (para conjuntos de dados com componentes discretos) obtidas pelos algoritmos. Nos resultados, também são mostrados os desvios padrão das médias das 10 iterações.

As Tabelas de resultados exibem as taxas médias de erro obtidas nos experimentos realizados com os sistemas híbridos de RBC e com os classificadores individuais. Os melhores modelos testados são apresentados em *itálico*. A coluna nomeada *k* representa o número de casos similares considerados durante

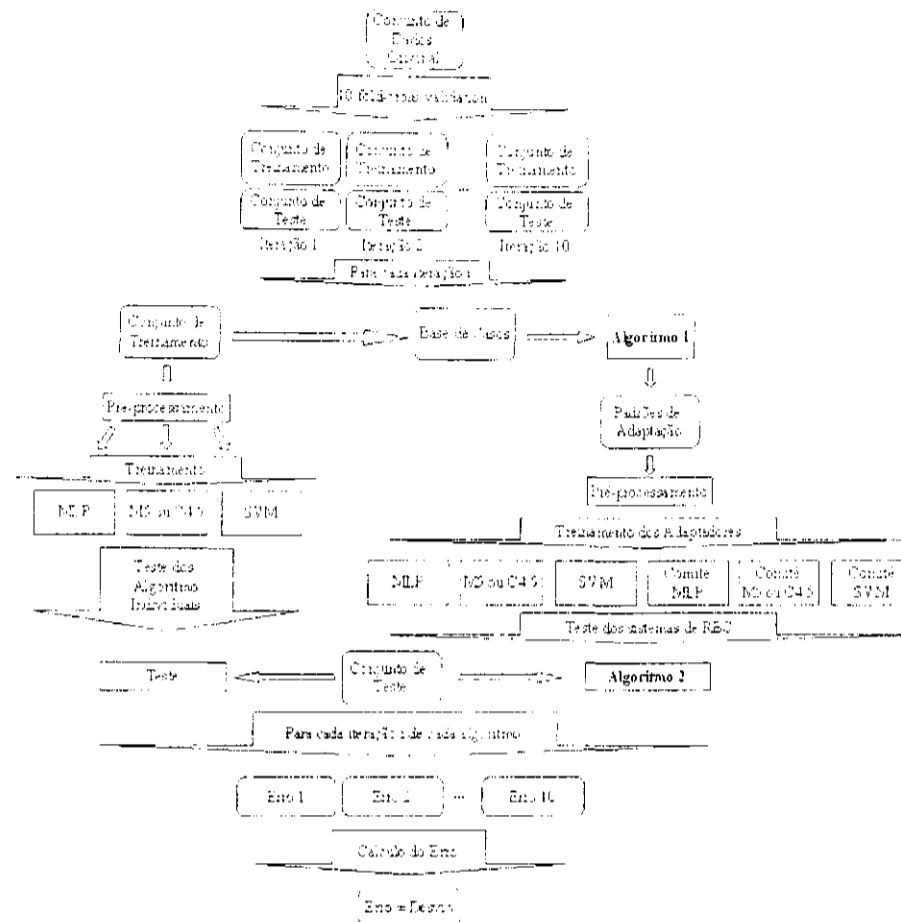


Figura 6.5: Metodologia dos experimentos realizados.

a geração dos padrões de adaptação, pela variação do parâmetro *NumeroCasos* no Algoritmo 2. Adicionalmente, são mostrados os resultados obtidos pelos sistemas híbridos de RBC utilizando as duas abordagens de recuperação de casos apontadas na Seção 6.6 (Kd - Tree e abordagem híbrida). Os resultados obtidos pelas técnicas de classificadores individuais também são mostrados.

Para os conjuntos de dados de regressão, são apresentados os valores máximo, médio e mínimo do domínio do problema, de maneira a mostrar a importância das taxas de erro absoluto obtidas.

Nos resultados apresentados, a seguinte notação é utilizada para apresentação dos modelos de sistemas de RBC testados:

**RBC (AR - AD - k)** nessa notação, *AR* indica a abordagem utilizada para a recuperação de casos (KdTree ou ART2). *AD* indica a abordagem utilizada para a adaptação de casos (ComitêMLP, M5, C4.5, por exemplo). *k* indica o número de casos similares ao caso prova considerados durante a geração dos padrões de adaptação. A utilização da notação RBC (KdTree-ComitêMLP-3), por exemplo, indica um modelo de sistema de RBC composto pela abordagem de recuperação de casos *Kd - Tree* e

pela abordagem de adaptação utilizando um comitê com o algoritmo MLP como combinador.

A seguir, serão apresentados os resultados obtidos para cada conjunto de dados.

### *Conjunto de dados Servo*

A Tabela 6.11 na página 86 mostra os resultados dos experimentos realizados para o conjunto de dados *Servo*.

Os resultados obtidos para este conjunto de dados mostram que as abordagens híbridas de adaptação de casos apresentam uma melhor predição da solução do problema em comparação aos classificadores utilizadas individualmente. Esses resultados sugerem que o conjunto de padrões de adaptação extraído da BC contém uma boa representação das adaptações necessárias sobre os componentes da solução do problema sendo resolvido.

Uma comparação entre os modelos híbridos de RBC utilizando algoritmos de AM individualmente como mecanismo de adaptação de casos e os modelos híbridos de RBC utilizando comitês – considerando os modelos que utilizam os mesmos algoritmos de AM como combinadores dos comitês ou como mecanismos individuais de adaptação de casos e considerando o mesmo valor de  $k$  para os modelos comparados, por exemplo RBC (ART2 - SVM - 5) e RBC (ART2 - ComitêSVM - 5) – mostra que os comitês melhoram o desempenho dos modelos híbridos de RBC.

Adicionalmente, os resultados mostram que, de uma maneira geral, o aumento do número de casos similares considerados durante a geração dos padrões de adaptação (coluna  $k$  das Tabelas de resultado) melhora o desempenho dos modelos híbridos de RBC. Além disso, os resultados mostram que a abordagem de recuperação de casos  $Kd - Tree$  possibilita, em geral, um melhor desempenho dos sistemas híbridos de RBC. Finalmente, os resultados mostram que modelo RBC(KdTree - ComitêMLP - 5) apresenta desempenho superior ou similar aos outros modelos testados.

Para mostrar a importância das taxas de erro obtidas, a Tabela 6.12 na página 87 exibe os valores do domínio das soluções para o conjunto de dados *Servo*.

Para confirmar o desempenho das abordagens investigadas, foi utilizado o Teste  $t$  pareado para procedimentos bilaterais com 99% de certeza (Moses, 1986; Mitchell, 1997a). Os resultados são mostrados na Tabela 6.13 na página 87.



### *Conjunto de dados Imports*

Embora os resultados obtidos para este conjunto de dados, apresentados na Tabela 6.14 na página 88, mostrem que algumas abordagens híbridas de RBC apresentam desempenho inferior às técnicas de classificadores utilizadas individualmente, os modelos de RBC híbridos utilizando comitês como mecanismo de adaptação de casos apresentam desempenho superior aos classificadores individuais.

Uma comparação entre os modelos híbridos de RBC utilizando comitês e os modelos híbridos de RBC utilizando os algoritmos de AM individualmente – semelhante à comparação realizada para os resultados obtidos para o conjunto de dados *Servo* – mostra que, em geral, os comitês melhoram o desempenho da maioria dos modelos híbridos de RBC.

Adicionalmente, os resultados mostram que a maioria dos modelos híbridos de RBC melhoram seu desempenho quando utilizam o valor de  $k$  igual a 3 durante a geração dos padrões de adaptação. Além disso, os resultados mostram que a abordagem de recuperação de casos *Kd - Tree* possibilita, em geral, um melhor desempenho dos sistemas híbridos de RBC. Finalmente, os resultados mostram que modelo RBC(KdTree - ComitêM5 - 3) apresenta desempenho superior ou similar aos outros modelos testados.

O conjunto de dados *Imports* original contém 205 casos. Porém, alguns casos apresentam valores ausentes para alguns atributos. Após a remoção dos casos com valores ausentes, esse conjunto de foi reduzido a 159 casos.

Para mostrar a importância das taxas de erro obtidas, a Tabela 6.15 na página 89 exibe os valores do domínio das soluções para o conjunto de dados *Imports*.

Para confirmar o desempenho das abordagens investigadas, foi utilizado o Teste  $t$  pareado para procedimentos bilaterais com 99% de certeza (Moses, 1986; Mitchell, 1997a). Os resultados são mostrados na Tabela 6.16 na página 89.

### *Conjunto de dados Housing*

A Tabela 6.17 na página 90 mostra os resultados dos experimento realizados para o conjunto de dados *Housing*.

Para este conjunto de dados foram obtidos resultados similares aos obtidos para o conjunto de dados *Imports*. Os resultados obtidos para este conjunto de dados mostram que os modelos híbridos de RBC utilizando comitês com algoritmos SVM como combinadores apresentam desempenho superior aos classificadores individuais.

A mesma comparação realizada entre os modelos híbridos de RBC utili-

zando comitês e os modelos híbridos de RBC utilizando os algoritmos de AM individualmente mostra que os comitês melhoram o desempenho da maioria dos modelos híbridos de RBC.

Adicionalmente, os resultados mostram que, em geral, o aumento do número de casos similares considerados durante a geração dos padrões de adaptação (coluna  $k$  das Tabelas de resultado) melhora o desempenho dos modelos híbridos de RBC. Além disso, os resultados mostram que as duas abordagens de recuperação de casos (*Kd - Tree* e abordagem híbrida) produziram resultados similares. Finalmente, os resultados mostram que modelo RBC(ART2 - ComitêSVM - 5) apresenta desempenho superior ou similar ao obtido pelos outros modelos testados.

Para mostrar a importância das taxas de erro obtidas, a Tabela 6.18 na página 91 exibe os valores do domínio das soluções para o conjunto de dados *Housing*.

Para confirmar o desempenho das abordagens investigadas, foi utilizado o Teste  $t$  pareado para procedimentos bilaterais com 99% de certeza (Moses, 1986; Mitchell, 1997a). Os resultados são mostrados na Tabela 6.19 na página 91.

### *Conjunto de dados Bridges*

Para este conjunto de dados, foram obtidos resultados similares aos obtidos para o conjunto de dados *Housing*. A média global dos erros obtidos para os cinco componentes de saída (*T - ou - D*, *Material*, *Espaçamento*, *Tamanho - Relativo* e *Tipo*) mostra que os modelos híbridos de RBC utilizando comitês com algoritmos SVM como combinadores apresentam desempenho superior aos classificadores individuais.

A mesma comparação realizada entre os modelos híbridos de RBC utilizando comitês e os modelos híbridos de RBC utilizando os algoritmos de AM individualmente mostra que os comitês melhoram o desempenho da maioria dos modelos híbridos de RBC.

Adicionalmente, os resultados mostram que a maioria dos modelos híbridos de RBC melhoram seu desempenho quando utilizam o valor de  $k$  igual a 3 durante a geração dos padrões de adaptação. Além disso, os resultados mostram que a abordagem híbrida de recuperação de casos possibilita, em geral, um melhor desempenho dos sistemas híbridos de RBC. Finalmente, a média global dos resultados para todos os componentes das soluções desse domínio mostra que o modelo RBC(ART2 - ComitêSVM - 5) apresenta desempenho superior ou similar aos outros modelos testados.

O conjunto de dados *Bridges* original contém 108 casos. Porém, alguns casos apresentam valores ausentes para alguns componentes da solução. Após

remover os casos com valores ausentes na solução, este conjunto foi reduzido a 89 casos. Para os experimento com este conjunto de dados foram aplicadas duas técnicas de pré-processamento de dados: a técnica para o preenchimento de valores ausentes proposta em (Batista, 2003) e a técnica para o balanceamento de conjunto de dados proposta em (Chawla et al., 2002). Embora uma discussão sobre técnicas de pré-processamento esteja além do escopo desta Dissertação de Mestrado, vale ressaltar que nos experimentos utilizando este conjunto de dados foram comparados os resultados obtidos pelos algoritmos após a aplicação das técnicas de pré-processamento apontadas acima e os resultados obtidos sem a aplicação das técnicas de pré-processamento. Os resultados mais promissores para este conjunto de dados foram os obtidos após a aplicação das técnicas de pré-processamento discutidas. Porém, esses resultados não são suficiente para concluir que essas técnicas são superiores a outras técnicas de pré-processamento existentes, exigindo uma experimentação mais detalhada com a aplicação dessas técnicas sobre um número maior de conjuntos de dados.

A Tabela 6.20 na página 92 mostra os resultados dos experimento realizados para o conjunto de dados *Bridges*.

Para confirmar o desempenho das abordagens investigadas, foi utilizado o Teste *t* pareado para procedimentos bilaterais com 99% de certeza (Moses, 1986; Mitchell, 1997a). Os resultados são mostrados na Tabela 6.21 na página 93.

#### *Conjunto de dados Water*

A Tabela 6.22 na página 94 mostra os resultados dos experimento realizados para o conjunto de dados *Water*.

Para este conjunto de dados, foram obtidos resultados similares aos obtidos para o conjunto de dados *Housing*. A média global dos erros obtidos para os sete componentes de saída (*Algal1*, *Algal2*, *Algal3*, *Algal4*, *Algal5*, *Algal6*, *Algal7*) mostra que os modelos híbridos de RBC, utilizando a abordagem híbrida para a recuperação de casos e os comitês com algoritmos SVM como combinadores como mecanismos de adaptação de casos, apresentam desempenho superior aos classificadores individuais.

A mesma comparação realizada entre os modelos híbridos de RBC utilizando comitês e os modelos híbridos de RBC utilizando os algoritmos de AM individualmente mostra que os comitês melhoram o desempenho da maioria dos modelos híbridos de RBC.

Adicionalmente, os resultados mostram que o aumento do número de casos similares considerados durante a geração dos padrões de adaptação (coluna *k* das Tabelas de resultado) contribui negativamente para o desempenho da mai-

oria dos modelos híbridos de RBC, exceto para os modelos híbridos de RBC utilizando o algoritmo SVM individualmente como mecanismo de adaptação e para os modelos utilizando os comitês com o algoritmo SVM como combinadores. Além disso, os resultados mostram que a abordagem de recuperação de casos *Kd-Tree* possibilita, em geral, um melhor desempenho dos sistemas híbridos de RBC. Finalmente, a média global dos resultados para todos os componentes das soluções desse domínio mostra que o modelo RBC(ART2 - ComitêSVM - 3) apresenta desempenho superior ou similar aos outros modelos testados.

O conjunto de dados *Water* original contém 340 casos. Porém, alguns casos apresentam valores ausentes para alguns atributos. Após a remoção dos casos com valores ausentes, esse conjunto de dados foi reduzido a 282 casos.

Para mostrar a importância das taxas de erro obtidas, a Tabela 6.23 na página 95 exibe os valores do domínio das soluções para o conjunto de dados *Water*.

Para confirmar o desempenho das abordagens investigadas, foi utilizado o Teste *t* pareado para procedimentos bilaterais com 99% de certeza (Moses, 1986; Mitchell, 1997a). Os resultados são mostrados na Tabela 6.24 na página 95.

### *Conjunto de dados Vinho*

Para este conjunto de dados, foram obtidos resultados similares aos obtidos para o conjunto de dados *Water*. A média global dos erros obtidos para os três componentes de saída (*PH*, *Índice*, *Nota*) mostra que os modelos híbridos de RBC, utilizando a abordagem híbrida para a recuperação de casos e os comitês com algoritmos SVM como combinadores como mecanismos de adaptação de casos, apresentam desempenho superior aos classificadores individuais.

A mesma comparação realizada entre os modelos híbridos de RBC utilizando comitês e os modelos híbridos de RBC utilizando os algoritmos de AM individualmente mostra que os comitês melhoram o desempenho da maioria dos modelos híbridos de RBC.

Adicionalmente, os resultados mostram que o aumento do número de casos similares considerados durante a geração dos padrões de adaptação (coluna *k* das Tabelas de resultado) contribui negativamente para o desempenho da maioria dos modelos híbridos de RBC, exceto para os modelos híbridos de RBC utilizando o algoritmo SVM individualmente como mecanismo de adaptação e utilizando os comitês com o algoritmo SVM como combinadores. Além disso, os resultados mostram que a abordagem de recuperação de casos *Kd-Tree* possibilita, em geral, um melhor desempenho dos sistemas híbridos de RBC.

### 6.8 Análise dos Resultados

Finalmente, a média global dos resultados para todos os componentes das soluções desse domínio mostra que o modelo RBC(KdTree - ComitêSVM - 5) apresenta desempenho superior ou similar aos outros modelos testados.

A Tabela 6.25 na página 96 mostra os resultados dos experimento realizados para o conjunto de dados *Vinho*.

Para mostrar a importância das taxas de erro obtidas, a Tabela 6.26 na página 97 exibe os valores do domínio das soluções para o conjunto de dados *Vinho*.

Para confirmar o desempenho das abordagens investigadas, foi utilizado o Teste *t* pareado para procedimentos bilaterais com 99% de certeza (Moses, 1986; Mitchell, 1997a). Os resultados são mostrados na Tabela 6.27 na página 97.

### 6.8 Análise dos Resultados

A Tabela 6.10 resume os resultados obtidos nos experimentos realizados, mostrando o modelo que obteve o melhor desempenho sobre cada conjunto dados utilizado, juntamente com algumas características do respectivo conjunto.

Tabela 6.10: Resultados obtidos.

Conjunto de Dados	#Exemplos	#Atributos (Contínuo, Simbólico)	#Atributos Ausentes	#Duplicados ou Conflitantes	Melhor Modelo
Servo	167	4(0,4)	0	0	RBC(KdTree-ComitêMLP-5)
Imports	205	25(14,11)	46	8	RBC(KdTree-ComitêM5-3)
Housing	506	13(13,0)	0	0	RBC(ART2-ComitêSVM-5)
Bridges	108	12(2,10)	38	0	RBC(ART2-ComitêSVM-5)
Water	340	18(15,3)	58	0	RBC(ART2-ComitêSVM-3)
Vinhos	289	13(12,1)	0	2	RBC(Kdtree-ComitêSVM-5)

Os resultados obtidos pelos experimentos realizados neste trabalho mostram que, em geral, os modelos híbridos de sistemas de RBC utilizando como mecanismos de adaptação de casos os comitês compostos pelo algoritmo SVM como combinador apresentam uma melhor predição da solução do problema em comparação as técnicas de classificadores utilizadas individualmente. Uma possível explicação para esses resultados pode estar nas propriedades do algoritmo SVM. Esse algoritmo emprega um controle de generalização e utiliza, para os experimentos realizados, o núcleo RBF para a construção de hiperplanos que formam regiões de decisão mais eficientes.

Os resultados obtidos indicam o potencial da combinação do Aprendizado Baseado em Instâncias com Aprendizado Indutivo e sugerem que o conjunto de padrões de adaptação extraído da BC contém uma boa representação das adaptações necessárias sobre os componentes das soluções do problema sendo resolvido.

Os resultados mostram também que, em geral, o aumento do número de casos similares ao caso prova considerados durante a geração dos padrões de adaptação (coluna  $k$  das Tabelas de resultado) melhora o desempenho dos modelos híbridos de RBC. Isso pode ser explicado pelo aumento da variabilidade das regras de adaptação codificadas nos padrões de adaptação. Com o aumento do número de casos similares considerados durante a geração dos padrões de adaptação, ocorre um aumento na cobertura das possíveis mudanças que os valores do domínio podem sofrer. Os resultados mostram ainda que, para a maioria dos modelos testados, a utilização de  $k$  igual a 3, isto é, a consideração de 3 casos similares ao caso prova durante a geração dos padrões de adaptação, produz melhores resultados.

Além disso, os resultados mostram que o mecanismo de recuperação de casos utilizado pelo sistema pode influenciar significativamente os resultados finais do sistema híbrido de RBC, sugerindo que a utilização de mecanismos de recuperação mais eficientes pode influenciar positivamente no desempenho dos sistemas de RBC.

Adicionalmente, os resultados mostram que, em geral, os comitês de algoritmos de AM introduzem mais acurácia e precisão ao sistema, isto é, os resultados mostram que as taxas de erro médias e os desvios padrão obtidos para os modelos utilizando comitês são, em geral, menores que os obtidos para os modelos híbridos de RBC utilizando os algoritmos de AM individualmente como mecanismo de adaptação de casos.

## 6.9 Considerações Finais

Este Capítulo apresentou as abordagens híbridas para adaptação de casos investigadas nesta Dissertação de Mestrado e os resultados dos experimentos realizados neste trabalho.

Os resultados mostram que os comitês de algoritmos de AM introduzem, para a maioria dos modelos testados, mais acurácia e precisão ao sistema, reduzindo a taxa do erro médio e o desvio padrão dos erros.

As abordagens de adaptação investigadas podem facilmente ser estendidas para domínios nos quais as soluções dos casos são compostas por mais de um componente, pela criação de conjuntos de dados de padrões de adaptação e de mecanismos de adaptação independentes para cada componente da solução e

tratando estes componentes como problemas distintos.

Essas abordagens de adaptação de casos empregam um processo de geração de padrões de adaptação que pode reduzir o esforço necessário para a aquisição de conhecimento necessário para a adaptação de casos para domínios que exijam um processo de adaptação por substituição. Em domínios com soluções compostas por mais de um componente pode ser necessário adquirir conhecimento somente para validar as soluções propostas pelo mecanismo de adaptação.

Além disso, o processo de geração de padrões de adaptação e o processo de adaptação de casos são totalmente integrados ao mecanismo de recuperação de casos e podem ser implementado empregando qualquer mecanismo usual de recuperação de casos.

Os resultados obtidos sugerem que o conjunto de padrões de adaptação extraído da BC contém uma boa representação das adaptações necessárias sobre os componentes da solução do problema sendo resolvido.

Tabela 6.11: Médias dos Erros Absolutos Obtidas para o conjunto de dados Servo.

Modelo	k	Média do Erro Absoluto	
ART2	RBC (M5)	1	0.032 ± 0.018
	RBC (M5)	3	0.028 ± 0.019
	RBC (M5)	5	0.025 ± 0.016
	RBC (MLP)	1	0.053 ± 0.022
	RBC (MLP)	3	0.034 ± 0.019
	RBC (MLP)	5	0.028 ± 0.015
	RBC (SVM)	1	0.050 ± 0.020
	RBC (SVM)	3	0.076 ± 0.010
	RBC (SVM)	5	0.075 ± 0.009
	RBC (ComitêM5)	1	0.032 ± 0.019
	RBC (ComitêM5)	3	0.032 ± 0.018
	RBC (ComitêM5)	5	0.032 ± 0.019
	RBC (ComitêMLP)	1	0.045 ± 0.020
	RBC (ComitêMLP)	3	0.028 ± 0.017
	RBC (ComitêMLP)	5	0.026 ± 0.014
	RBC (ComitêSVM)	1	0.035 ± 0.015
	RBC (ComitêSVM)	3	0.039 ± 0.017
	RBC (ComitêSVM)	5	0.036 ± 0.015
Kd-Tree	RBC (M5)	1	0.032 ± 0.017
	RBC (M5)	3	0.026 ± 0.018
	RBC (M5)	5	0.024 ± 0.019
	RBC (MLP)	1	0.033 ± 0.020
	RBC (MLP)	3	0.033 ± 0.020
	RBC (MLP)	5	0.034 ± 0.019
	RBC (SVM)	1	0.078 ± 0.010
	RBC (SVM)	3	0.077 ± 0.009
	RBC (SVM)	5	0.075 ± 0.010
	RBC (ComitêM5)	1	0.032 ± 0.019
	RBC (ComitêM5)	3	0.032 ± 0.018
	RBC (ComitêM5)	5	0.032 ± 0.019
	RBC (ComitêMLP)	1	0.033 ± 0.019
	RBC (ComitêMLP)	3	0.030 ± 0.020
	RBC (ComitêMLP)	5	0.024 ± 0.018
	RBC (ComitêSVM)	1	0.042 ± 0.014
	RBC (ComitêSVM)	3	0.038 ± 0.015
	RBC (ComitêSVM)	5	0.043 ± 0.019
M5		0.976 ± 0.302	
MLP		0.959 ± 0.431	
SVM		0.832 ± 0.319	



Tabela 6.12: Valores do Domínio para o Conjunto de Dados Servo.

Descrição	Valor
valor Máximo	7.10
Valor Mínimo	0.13
Valor Médio	$1.41 \pm 1.58$

Tabela 6.13: Resultados do Teste  $t$  para o conjunto de dados Servo.

Modelos Comparados	Conclusão
RBC (KdTree - M5 - 5) e RBC (KdTree - ComitêMLP - 5)	Desempenho Similar
RBC (KdTree - ComitêMLP - 5) e MLP	RBC (KdTree - ComitêMLP - 5) é superior
RBC (KdTree - ComitêMLP - 5) e M5	RBC (KdTree - ComitêMLP - 5) é superior
RBC (KdTree - ComitêMLP - 5) e SVM	RBC (KdTree - ComitêMLP - 5) é superior

Tabela 6.14: Médias dos Erros Absolutos Obtidas para o conjunto de dados Imports.

Modelo	k	Média do Erro Absoluto	
ART2	RBC (M5)	1	1117.78 ± 272.86
	RBC (M5)	3	604.07 ± 77.26
	RBC (M5)	5	644.07 ± 82.47
	RBC (MLP)	1	1939.88 ± 649.63
	RBC (MLP)	3	1028.35 ± 136.36
	RBC (MLP)	5	998.35 ± 127.67
	RBC (SVM)	1	1415.33 ± 230.26
	RBC (SVM)	3	1078.35 ± 95.73
	RBC (SVM)	5	1088.35 ± 67.69
	RBC (ComitêM5)	1	1025.42 ± 217.58
	<i>RBC(ComitêM5)</i>	3	576.85 ± 83.05
	RBC (ComitêM5)	5	734.07 ± 95.28
	RBC (ComitêMLP)	1	1157.00 ± 177.14
	RBC (ComitêMLP)	3	626.19 ± 71.25
	RBC (ComitêMLP)	5	696.19 ± 89.59
	RBC (ComitêSVM)	1	1107.90 ± 135.84
	RBC (ComitêSVM)	3	616.19 ± 71.25
	RBC (ComitêSVM)	5	636.19 ± 50.58
Kd-Tree	RBC (M5)	1	1127.78 ± 266.24
	RBC (M5)	3	604.07 ± 77.26
	RBC (M5)	5	654.07 ± 71.47
	RBC (MLP)	1	1835.88 ± 462.89
	RBC (MLP)	3	1038.35 ± 130.25
	RBC (MLP)	5	998.35 ± 117.59
	RBC (SVM)	1	1395.33 ± 213.79
	RBC (SVM)	3	1087.34 ± 108.34
	RBC (SVM)	5	1077.35 ± 69.67
	RBC (ComitêM5)	1	1005.42 ± 195.30
	<i>RBC(ComitêM5)</i>	3	576.05 ± 82.58
	RBC (ComitêM5)	5	725.04 ± 87.20
	RBC (ComitêMLP)	1	1168.00 ± 169.42
	RBC (ComitêMLP)	3	636.16 ± 86.90
	RBC (ComitêMLP)	5	685.21 ± 91.81
	RBC (ComitêSVM)	1	1184.00 ± 111.22
	RBC (ComitêSVM)	3	627.19 ± 67.09
	RBC (ComitêSVM)	5	628.10 ± 65.09
M5		1603.66 ± 339.53	
MLP		1761.31 ± 483.25	
SVM		1661.31 ± 470.39	

Tabela 6.15: Valores do Domínio para o Conjunto de Dados Imports.

Descrição	Valor
valor Máximo	35050.00
Valor Mínimo	5118.00
Valor Médio	11415.73 ± 5877.86

Tabela 6.16: Resultados do Teste  $t$  para o conjunto de dados Imports.

Modelos Comparados	Conclusão
RBC (ART2 - ComitêM5 - 3) e RBC (KdTree - ComitêM5 - 3)	Desempenho Similar
RBC (KdTree - ComitêM5 - 3) e MLP	RBC (KdTree - ComitêM5 - 3) é superior
RBC (KdTree - ComitêM5 - 3) e M5	RBC (KdTree - ComitêM5 - 3) é superior
RBC (KdTree - ComitêM5 - 3) e SVM	RBC (KdTree - ComitêM5 - 3) é superior

Tabela 6.17: Médias dos Erros Absolutos Obtidas para o conjunto de dados Housing.

Modelo	k	Média do Erro Absoluto	
ART2	RBC (M5)	1	3.45 ± 0.28
	RBC (M5)	3	3.78 ± 0.51
	RBC (M5)	5	4.10 ± 0.64
	RBC (MLP)	1	3.94 ± 0.52
	RBC (MLP)	3	3.96 ± 0.46
	RBC (MLP)	5	4.00 ± 0.38
	RBC (SVM)	1	3.46 ± 0.38
	RBC (SVM)	3	3.17 ± 0.31
	RBC (SVM)	5	3.11 ± 0.29
	RBC (ComitêM5)	1	3.47 ± 0.36
	RBC (ComitêM5)	3	3.53 ± 0.40
	RBC (ComitêM5)	5	3.95 ± 0.90
	RBC (ComitêMLP)	1	3.76 ± 0.46
	RBC (ComitêMLP)	3	3.67 ± 0.36
	RBC (ComitêMLP)	5	3.63 ± 0.34
	RBC (ComitêSVM)	1	3.07 ± 1.63
	RBC (ComitêSVM)	3	2.44 ± 0.35
	RBC (ComitêSVM)	5	2.37 ± 0.46
Kd-Tree	RBC (M5)	1	3.43 ± 0.29
	RBC (M5)	3	3.77 ± 0.45
	RBC (M5)	5	3.44 ± 0.45
	RBC (MLP)	1	4.51 ± 0.66
	RBC (MLP)	3	4.00 ± 0.58
	RBC (MLP)	5	4.44 ± 0.54
	RBC (SVM)	1	3.45 ± 0.35
	RBC (SVM)	3	3.19 ± 0.19
	RBC (SVM)	5	3.12 ± 0.28
	RBC (ComitêM5)	1	3.77 ± 0.45
	RBC (ComitêM5)	3	3.55 ± 0.51
	RBC (ComitêM5)	5	4.42 ± 1.17
	RBC (ComitêMLP)	1	3.85 ± 0.35
	RBC (ComitêMLP)	3	3.44 ± 0.45
	RBC (ComitêMLP)	5	3.87 ± 0.48
	RBC (ComitêSVM)	1	2.78 ± 0.12
	RBC (ComitêSVM)	3	2.63 ± 0.12
	RBC (ComitêSVM)	5	2.62 ± 0.17
M5		3.43 ± 0.29	
MLP		4.28 ± 0.94	
SVM		3.18 ± 0.33	

Tabela 6.18: Valores do Domínio para o Conjunto de Dados Housing.

Descrição	Valor
valor Máximo	50.00
Valor Mínimo	5.00
Valor Médio	22.53 ± 9.20

Tabela 6.19: Resultados do Teste  $t$  para o conjunto de dados Housing.

Modelos Comparados	Conclusão
RBC (ART2 - ComitêSVM - 5) e RBC (ART2 - ComitêSVM - 3)	RBC (ART2 - ComitêSVM - 5) é superior
RBC (ART2 - ComitêSVM - 5) e MLP	RBC (ART2 - ComitêSVM - 5) é superior
RBC (ART2 - ComitêSVM - 5) e M5	RBC (ART2 - ComitêSVM - 5) é superior
RBC (ART2 - ComitêSVM - 5) e SVM	RBC (ART2 - ComitêSVM - 5) é superior

Tabela 6.20: Médias dos Erros (%) Obtidas para o conjunto de dados Bridges.

Modelo		K	Média do Erro de Classificação (%)				
			t-ou-d	tam-rel	espaçamento	material	tipo
ART2	RBC (C4.5)	1	31.11 ± 9.02	65.24 ± 9.98	45.74 ± 7.98	32.30 ± 6.95	53.19 ± 8.02
	RBC (C4.5)	3	28.39 ± 7.07	54.27 ± 11.06	50.27 ± 8.05	33.03 ± 3.54	53.18 ± 9.98
	RBC (C4.5)	5	31.11 ± 8.34	55.17 ± 9.00	52.17 ± 7.21	35.03 ± 5.43	56.09 ± 11.50
	RBC (SVM)	1	16.00 ± 7.73	31.33 ± 9.59	41.22 ± 9.25	24.22 ± 5.51	48.22 ± 13.70
	RBC(SVM)	3	6.00 ± 5.59	31.33 ± 9.59	36.22 ± 10.58	13.22 ± 6.77	47.22 ± 12.21
	RBC (SVM)	5	6.00 ± 5.59	50.89 ± 15.56	36.22 ± 10.58	10.00 ± 5.33	46.22 ± 13.41
	RBC (MLP)	1	23.84 ± 4.97	55.27 ± 12.76	45.74 ± 7.98	34.12 ± 6.69	56.99 ± 7.32
	RBC (MLP)	3	21.75 ± 4.77	56.08 ± 7.88	50.27 ± 8.05	29.39 ± 4.58	57.71 ± 8.03
	RBC (MLP)	5	24.75 ± 4.77	49.74 ± 10.57	52.17 ± 7.21	31.12 ± 4.78	55.80 ± 8.85
	RBC (ComitêC4.5)	1	23.84 ± 4.97	55.27 ± 12.76	43.11 ± 7.52	34.12 ± 6.69	59.53 ± 12.65
	RBC (ComitêC4.5)	3	28.39 ± 7.07	54.27 ± 11.06	50.27 ± 8.05	33.03 ± 5.54	53.18 ± 9.98
	RBC (ComitêC4.5)	5	31.11 ± 8.34	55.17 ± 9.00	51.26 ± 7.21	35.03 ± 5.43	56.09 ± 11.50
	RBC (ComitêMLP)	1	23.84 ± 3.60	55.27 ± 12.76	44.01 ± 8.75	34.12 ± 6.69	62.25 ± 12.37
	RBC (ComitêMLP)	3	25.66 ± 5.60	54.27 ± 11.06	50.27 ± 8.05	30.21 ± 5.75	53.08 ± 12.41
	RBC (ComitêMLP)	5	26.57 ± 7.89	49.74 ± 10.57	51.26 ± 5.54	31.12 ± 4.78	54.09 ± 10.88
	RBC (ComitêSVM)	1	6.00 ± 5.59	43.67 ± 16.17	35.33 ± 10.07	17.33 ± 8.87	46.44 ± 6.56
	RBC (ComitêSVM)	3	7.00 ± 5.40	42.56 ± 10.25	36.33 ± 10.70	13.22 ± 6.77	51.33 ± 11.81
	RBC(ComitêSVM)	5	7.00 ± 5.40	32.33 ± 7.34	36.67 ± 11.31	10.00 ± 5.33	48.22 ± 11.44
Kd-Tree	RBC (C4.5)	1	32.02 ± 10.82	58.81 ± 10.74	45.74 ± 7.98	34.12 ± 6.69	55.99 ± 8.75
	RBC (C4.5)	3	31.12 ± 6.81	58.71 ± 10.89	51.17 ± 9.59	31.30 ± 5.43	56.50 ± 11.48
	RBC (C4.5)	5	31.12 ± 6.81	56.72 ± 10.66	52.16 ± 7.98	32.21 ± 6.44	55.00 ± 11.33
	RBC (SVM)	1	29.11 ± 6.42	35.22 ± 9.94	39.22 ± 11.35	20.22 ± 5.36	49.44 ± 7.67
	RBC (SVM)	3	18.00 ± 8.26	34.33 ± 10.74	39.22 ± 11.35	26.33 ± 6.93	57.56 ± 9.08
	RBC (SVM)	5	18.00 ± 8.26	32.33 ± 10.52	34.22 ± 9.17	27.44 ± 8.18	52.56 ± 7.38
	RBC (MLP)	1	41.19 ± 13.55	59.98 ± 6.71	58.35 ± 9.46	48.47 ± 10.74	67.80 ± 4.39
	RBC (MLP)	3	38.56 ± 13.06	61.24 ± 6.66	67.88 ± 7.78	49.55 ± 11.91	63.80 ± 10.11
	RBC (MLP)	5	44.19 ± 14.81	57.16 ± 4.39	58.72 ± 9.13	48.01 ± 10.98	66.17 ± 9.98
	RBC (ComitêC4.5)	1	40.28 ± 13.96	59.98 ± 6.71	58.35 ± 9.43	46.47 ± 9.56	59.53 ± 12.65
	RBC (ComitêC4.5)	3	31.12 ± 6.81	58.71 ± 10.89	51.17 ± 9.95	31.75 ± 5.84	53.18 ± 9.98
	RBC (ComitêC4.5)	5	31.12 ± 6.81	56.72 ± 10.66	52.16 ± 7.98	32.21 ± 6.44	56.09 ± 11.50
	RBC (ComitêMLP)	1	31.74 ± 13.72	61.89 ± 6.96	58.35 ± 9.43	43.57 ± 10.44	68.34 ± 5.28
	RBC (ComitêMLP)	3	29.30 ± 5.51	57.80 ± 10.28	52.08 ± 9.23	30.39 ± 5.31	68.34 ± 5.28
	RBC (ComitêMLP)	5	27.48 ± 4.63	56.72 ± 10.66	52.16 ± 5.34	33.12 ± 6.35	53.00 ± 11.28
	RBC (ComitêSVM)	1	32.00 ± 10.53	50.56 ± 11.40	48.56 ± 12.54	39.00 ± 13.31	60.67 ± 10.12
	RBC (ComitêSVM)	3	28.00 ± 19.52	51.22 ± 13.29	54.56 ± 7.65	41.11 ± 19.64	62.67 ± 9.14
	RBC (ComitêSVM)	5	33.00 ± 12.54	48.44 ± 8.87	52.16 ± 5.34	39.00 ± 13.31	60.67 ± 9.39
	C4.5		28.38 ± 9.22	58.81 ± 10.74	45.74 ± 7.98	31.39 ± 6.95	59.69 ± 7.54
	SVM		19.00 ± 7.96	31.33 ± 10.30	39.22 ± 11.35	24.22 ± 5.51	57.44 ± 11.56
	MLP		25.66 ± 6.56	46.91 ± 8.48	45.74 ± 7.98	31.30 ± 6.43	64.35 ± 10.32

Tabela 6.21: Resultados do Teste  $t$  para o conjunto de dados Bridges.

Modelos Comparados	Conclusão
RBC (ART2 - SVM - 3) e RBC (ART2 - ComitêSVM - 5)	RBC (ART2 - ComitêSVM - 5) é superior
RBC (ART2 - ComitêSVM - 5) e MLP	RBC (ART2 - ComitêSVM - 5) é superior
RBC (ART2 - ComitêSVM - 5) e M5	RBC (ART2 - ComitêSVM - 5) é superior
RBC (ART2 - ComitêSVM - 5) e SVM	RBC (ART2 - ComitêSVM - 5) é superior

Tabela 6.22: Médias dos Erros Absolutos Obtidas para o conjunto de dados Water.

Modelo		K	Média do Erro Absoluto						
			Alga 1	Alga 2	Alga 3	Alga 4	Alga 5	Alga 6	Alga 7
ART2	RBC (M5)	1	9.86 ± 1.71	6.51 ± 0.65	4.31 ± 0.45	1.38 ± 0.17	5.24 ± 0.60	6.66 ± 0.86	2.38 ± 0.50
	RBC (M5)	3	9.56 ± 2.20	7.13 ± 0.91	4.40 ± 0.63	1.57 ± 0.25	5.24 ± 0.60	6.90 ± 1.15	2.62 ± 0.39
	RBC (M5)	5	10.42 ± 2.54	8.54 ± 1.74	4.64 ± 0.74	1.71 ± 0.25	6.29 ± 1.48	7.14 ± 1.42	2.64 ± 0.54
	RBC (SVM)	1	11.42 ± 0.85	7.55 ± 0.57	4.61 ± 0.43	1.11 ± 0.12	1.14 ± 0.12	7.22 ± 0.72	3.19 ± 0.25
	RBC (SVM)	3	11.30 ± 0.97	7.49 ± 0.62	4.57 ± 0.51	1.38 ± 0.14	5.61 ± 0.36	6.75 ± 0.80	3.11 ± 0.25
	RBC (SVM)	5	11.38 ± 1.05	7.35 ± 0.64	4.56 ± 0.53	1.37 ± 0.15	5.62 ± 0.40	6.79 ± 0.89	3.12 ± 0.26
	RBC (MLP)	1	27.19 ± 3.29	8.96 ± 1.33	10.66 ± 1.24	3.25 ± 0.53	11.61 ± 1.80	11.76 ± 1.11	5.86 ± 0.90
	RBC (MLP)	3	24.30 ± 5.39	7.86 ± 0.88	8.46 ± 0.81	3.01 ± 0.50	9.35 ± 1.03	1.23 ± 1.64	5.67 ± 0.83
	RBC (MLP)	5	26.49 ± 4.16	13.92 ± 1.35	8.94 ± 2.16	3.08 ± 0.52	10.25 ± 1.53	12.02 ± 1.62	5.77 ± 0.88
	RBC (ComiteM5)	1	11.11 ± 1.91	12.03 ± 2.19	7.69 ± 1.29	2.19 ± 0.49	9.48 ± 1.73	9.15 ± 0.75	2.38 ± 0.50
	RBC (ComiteM5)	3	14.56 ± 2.92	11.00 ± 1.46	5.64 ± 0.50	2.02 ± 0.36	7.23 ± 1.01	10.01 ± 1.26	2.62 ± 0.39
	RBC (ComiteM5)	5	15.98 ± 2.43	10.72 ± 1.55	6.30 ± 1.42	2.13 ± 0.42	7.75 ± 1.21	9.20 ± 1.16	2.64 ± 0.54
	RBC (ComiteMLP)	1	13.99 ± 2.25	11.82 ± 2.04	7.15 ± 1.15	2.19 ± 0.41	9.59 ± 1.41	10.08 ± 0.91	5.02 ± 0.73
	RBC (ComiteMLP)	3	16.19 ± 4.22	12.07 ± 1.34	6.24 ± 0.79	2.10 ± 0.42	8.07 ± 0.91	10.82 ± 1.22	4.23 ± 0.45
	RBC (ComiteMLP)	5	16.44 ± 2.11	12.29 ± 1.34	6.74 ± 0.95	2.53 ± 0.45	8.07 ± 0.92	10.68 ± 1.59	4.69 ± 0.77
	RBC (ComiteSVM)	1	5.05 ± 0.59	3.49 ± 0.15	1.86 ± 0.23	0.63 ± 0.07	2.84 ± 0.35	4.33 ± 0.21	1.50 ± 0.25
	RBC (ComiteSVM)	3	3.46 ± 0.67	3.77 ± 0.38	1.47 ± 0.26	0.53 ± 0.10	3.39 ± 0.30	3.80 ± 0.33	1.77 ± 0.17
	RBC (ComiteSVM)	5	3.34 ± 0.53	3.92 ± 0.15	1.93 ± 0.20	0.71 ± 0.08	3.48 ± 0.24	3.80 ± 0.40	1.56 ± 0.22
K4 Tree	RBC (M5)	1	6.63 ± 1.10	5.36 ± 1.13	3.16 ± 0.67	1.06 ± 0.22	4.51 ± 0.95	6.56 ± 0.83	2.27 ± 0.60
	RBC (M5)	3	8.18 ± 2.73	6.61 ± 2.20	3.90 ± 1.30	1.30 ± 0.43	5.17 ± 1.07	6.80 ± 1.40	2.42 ± 0.54
	RBC (M5)	5	9.49 ± 5.23	7.67 ± 4.22	4.52 ± 2.49	1.51 ± 0.83	5.25 ± 0.99	7.04 ± 1.30	2.53 ± 0.55
	RBC (SVM)	1	8.83 ± 0.65	7.14 ± 0.53	4.21 ± 0.31	1.11 ± 0.10	6.01 ± 0.45	7.12 ± 0.64	3.10 ± 0.36
	RBC (SVM)	3	8.73 ± 0.65	7.06 ± 0.52	4.16 ± 0.31	1.39 ± 0.10	5.94 ± 0.44	6.65 ± 0.75	2.94 ± 0.55
	RBC (SVM)	5	8.68 ± 0.61	7.01 ± 0.52	4.13 ± 0.31	1.38 ± 0.10	5.90 ± 0.44	6.69 ± 0.80	3.01 ± 0.47
	RBC (MLP)	1	18.61 ± 3.63	11.46 ± 4.24	8.52 ± 2.50	2.85 ± 0.81	12.17 ± 3.57	11.66 ± 1.09	5.75 ± 0.76
	RBC (MLP)	3	15.77 ± 2.90	13.20 ± 3.13	7.78 ± 1.85	2.60 ± 0.62	11.11 ± 2.64	12.13 ± 1.56	5.47 ± 0.81
	RBC (MLP)	5	15.66 ± 3.70	13.50 ± 3.50	8.21 ± 2.03	2.74 ± 0.68	12.11 ± 2.72	11.92 ± 1.51	5.66 ± 0.86
	RBC (ComiteM5)	1	3.70 ± 1.78	9.66 ± 3.02	6.29 ± 2.65	2.10 ± 0.88	8.97 ± 3.78	9.05 ± 0.70	4.66 ± 0.80
	RBC (ComiteM5)	3	11.35 ± 2.95	9.62 ± 3.07	5.67 ± 1.81	1.89 ± 0.60	8.10 ± 2.58	9.91 ± 1.26	3.67 ± 0.53
	RBC (ComiteM5)	5	11.30 ± 3.96	8.36 ± 2.75	5.47 ± 2.31	1.83 ± 0.78	8.04 ± 3.21	9.10 ± 1.21	3.96 ± 0.82
	RBC (ComiteMLP)	1	14.67 ± 4.09	10.75 ± 3.54	6.33 ± 2.27	2.12 ± 0.76	9.04 ± 3.24	9.98 ± 0.90	4.93 ± 0.71
	RBC (ComiteMLP)	3	12.72 ± 3.15	10.95 ± 3.14	6.46 ± 1.55	2.16 ± 0.62	9.22 ± 2.65	10.72 ± 1.26	4.03 ± 0.49
	RBC (ComiteMLP)	5	13.49 ± 4.13	11.10 ± 3.70	6.70 ± 2.17	2.21 ± 0.72	9.78 ± 2.97	10.58 ± 1.57	4.58 ± 0.85
	RBC (ComiteSVM)	1	4.62 ± 0.67	3.50 ± 0.70	2.07 ± 0.41	0.69 ± 0.14	2.95 ± 0.59	4.43 ± 0.27	1.61 ± 0.33
	RBC (ComiteSVM)	3	5.23 ± 0.58	4.27 ± 0.50	2.51 ± 0.29	0.84 ± 0.10	3.59 ± 0.42	3.90 ± 0.47	1.88 ± 0.24
	RBC (ComiteSVM)	5	4.85 ± 0.58	4.02 ± 0.44	2.39 ± 0.23	0.80 ± 0.08	3.39 ± 0.32	3.80 ± 0.40	1.67 ± 0.31
M5			11.23 ± 0.80	6.34 ± 0.59	4.43 ± 0.40	1.38 ± 0.18	5.03 ± 0.57	6.55 ± 0.57	2.34 ± .48
SVM			10.14 ± 1.26	7.39 ± 0.59	4.62 ± 0.41	1.40 ± 0.13	5.52 ± 0.41	6.86 ± 0.75	3.17 ± 0.24
MLP			21.05 ± 2.70	9.43 ± 1.47	9.98 ± 1.10	3.20 ± 0.62	12.41 ± 2.52	14.06 ± 2.52	5.86 ± 1.92



Tabela 6.23: Valores do Domínio para o Conjunto de Dados Water.

Descrição	Valor						
	Alga 1	Alga 2	Alga 3	Alga 4	Alga 5	Alga 6	Alga 7
valor Máximo	89.80	72.60	42.80	14.30	61.10	70.0	31.60
Valor Mínimo	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Valor Médio	15.13 ± 19.71	7.58 ± 11.03	4.45 ± 8.81	1.36 ± 2.22	6.12 ± 8.74	6.84 ± 12.13	2.24 ± 5.05

Tabela 6.24: Resultados do Teste *t* para o conjunto de dados Water.

Modelos Comparados	Conclusão
RBC (ART2 - ComitêSVM - 3) e RBC (ART2 - ComitêSVM - 5)	RBC (ART2 - ComitêSVM - 3) é superior
RBC (ART2 - ComitêSVM - 3) e MLP	RBC (ART2 - ComitêSVM - 3) é superior
RBC (ART2 - ComitêSVM - 3) e M5	RBC (ART2 - ComitêSVM - 3) é superior
RBC (ART2 - ComitêSVM - 3) e SVM	RBC (ART2 - ComitêSVM - 3) é superior

Tabela 6.25: Médias dos Erros Absolutos Obtidas para o conjunto de dados Vinho.

Modelo		K	Média do Erro Absoluto		
			índice	Nota	PH
ART2	RBC (M5)	1	12.03 ± 0.60	3.30 ± 0.42	0.21 ± 0.05
	RBC (M5)	3	12.28 ± 0.70	3.77 ± 0.60	0.22 ± 0.05
	RBC (M5)	5	12.96 ± 0.98	4.22 ± 0.80	0.27 ± 0.17
	RBC (SVM)	1	8.03 ± 0.47	3.01 ± 0.30	0.11 ± 0.03
	RBC (SVM)	3	7.38 ± 0.33	3.03 ± 0.29	0.16 ± 0.10
	RBC (SVM)	5	7.40 ± 0.32	3.11 ± 0.57	0.17 ± 0.12
	RBC (MLP)	1	7.75 ± 0.99	4.46 ± 0.90	0.21 ± 0.03
	RBC (MLP)	3	13.10 ± 1.01	4.46 ± 0.65	0.28 ± 0.20
	RBC (MLP)	5	13.20 ± 1.11	4.75 ± 0.76	0.25 ± 0.12
	RBC (ComiteM5)	1	11.65 ± 0.84	4.20 ± 0.87	0.21 ± 0.05
	RBC (ComiteM5)	3	12.23 ± 0.60	4.27 ± 0.78	0.22 ± 0.07
	RBC (ComiteM5)	5	12.78 ± 0.68	4.88 ± 0.16	0.28 ± 0.14
	RBC (ComiteMLP)	1	12.34 ± 0.61	4.15 ± 0.47	0.24 ± 0.09
	RBC (ComiteMLP)	3	12.17 ± 0.83	4.27 ± 0.57	0.20 ± 0.03
	RBC (ComiteMLP)	5	12.35 ± 0.84	4.55 ± 0.64	0.28 ± 0.20
	RBC (ComiteSVM)	1	5.60 ± 0.35	2.47 ± 0.48	0.09 ± 0.00
	RBC (ComiteSVM)	3	4.95 ± 0.41	1.75 ± 0.14	0.09 ± 0.06
	RBC (ComiteSVM)	5	3.64 ± 0.30	1.47 ± 0.45	0.07 ± 0.03
Kd-Tree	RBC (M5)	1	11.93 ± 0.66	3.16 ± 0.21	0.18 ± 0.01
	RBC (M5)	3	12.20 ± 0.67	3.59 ± 0.57	0.19 ± 0.02
	RBC (M5)	5	12.87 ± 1.03	4.07 ± 0.86	0.20 ± 0.02
	RBC (SVM)	1	7.90 ± 0.32	2.91 ± 0.16	0.12 ± 0.01
	RBC (SVM)	3	7.32 ± 0.29	2.91 ± 0.16	0.11 ± 0.01
	RBC (SVM)	5	7.36 ± 0.30	2.91 ± 0.16	0.11 ± 0.01
	RBC (MLP)	1	7.55 ± 1.21	4.25 ± 0.95	0.19 ± 0.02
	RBC (MLP)	3	13.17 ± 1.09	4.28 ± 0.76	0.20 ± 0.01
	RBC (MLP)	5	13.05 ± 1.92	4.63 ± 0.70	0.20 ± 0.02
	RBC (ComiteM5)	1	11.53 ± 0.85	4.08 ± 0.82	0.18 ± 0.01
	RBC (ComiteM5)	3	12.09 ± 0.66	4.10 ± 0.72	0.19 ± 0.01
	RBC (ComiteM5)	5	12.72 ± 0.68	4.84 ± 0.41	0.24 ± 0.07
	RBC (ComiteMLP)	1	12.31 ± 0.63	4.04 ± 0.58	0.19 ± 0.01
	RBC (ComiteMLP)	3	12.10 ± 0.86	4.13 ± 0.65	0.19 ± 0.01
	RBC (ComiteMLP)	5	12.13 ± 0.69	4.43 ± 0.66	0.20 ± 0.01
	RBC (ComiteSVM)	1	5.57 ± 0.31	2.31 ± 0.08	0.08 ± 0.00
	RBC (ComiteSVM)	3	3.86 ± 0.33	1.58 ± 0.07	0.06 ± 0.00
	RBC (ComiteSVM)	5	3.57 ± 0.31	1.29 ± 0.07	0.06 ± 0.00
	M5		12.00 ± 0.71	3.17 ± 0.21	0.19 ± 0.01
	SVM		8.05 ± 0.53	2.88 ± 0.15	0.13 ± 0.02
	MLP		8.71 ± 1.27	4.11 ± 0.63	0.24 ± 0.07

Tabela 6.26: Valores do Domínio para o Conjunto de Dados Vinho.

Descrição	Valor		
	Índice	Nota	PH
valor Máximo	93.10	88.00	4.09
Valor Mínimo	1.40	61.00	3.20
Valor Médio	42.44 ± 23.59	80.84 ± 3.85	3.67 ± 0.19

Tabela 6.27: Resultados do Teste  $t$  para o conjunto de dados Vinho.

Modelos Comparados	Conclusão
RBC (KdTree - ComitêSVM - 5) e RBC (ART2 - ComitêSVM - 5)	RBC (KdTree - ComitêSVM - 5) é superior
RBC (KdTree - ComitêSVM - 5) e MLP	RBC (KdTree - ComitêSVM - 5) é superior
RBC (KdTree - ComitêSVM - 5) e M5	RBC (KdTree - ComitêSVM - 5) é superior
RBC (KdTree - ComitêSVM - 5) e SVM	RBC (KdTree - ComitêSVM - 5) é superior



---

## Conclusão

---

### 7.1 *Considerações Iniciais*

Neste Capítulo são apresentadas as conclusões deste trabalho. Na Seção 7.3 são feitas as principais considerações sobre esta Dissertação de Mestrado e sobre os resultados obtidos. Na Seção 7.3 são apresentadas brevemente as possibilidades de trabalhos futuros. Na Seção 7.4 são listadas as publicações originadas deste trabalho.

### 7.2 *Discussão*

A adaptação de casos é um dos maiores desafios da área de RBC. Em contraste com a aquisição de casos, a aquisição de conhecimento necessário para a adaptação de casos exige um considerável esforço de aquisição (Hanney, 1996; Wiratunga et al., 2002). A forma de adaptação mais utilizada é a codificação de regras de adaptação, demandando um significativo esforço para a aquisição de conhecimento (Hanney, 1996).

A construção de abordagens de adaptação genéricas é tida como improvável (Kolodner, 1993), apesar de essa ainda ser uma área de pesquisa em aberto. As pesquisas atuais em adaptação de casos visam a redução do esforço de aquisição de conhecimento específico do domínio de aplicação.

Uma das possíveis alternativas para superar as dificuldades em adquirir o conhecimento necessário para realizar a adaptação de casos e para a construção de mecanismos de adaptação de casos que possam ser utilizados em vários domínios, é o desenvolvimento de abordagens que empregam outras técnicas da IA e que trabalham com um conjunto reduzido de conhecimento

do domínios, reduzindo o esforço de aquisição do conhecimento necessário.

As abordagens investigadas nesta Dissertação de Mestrado possibilitam uma redução no esforço de aquisição de conhecimento por meio da aquisição automática de padrões de adaptação diretamente de um repositório de conhecimento já adquirido: a BC. Em aplicações mais avançadas, a aquisição de conhecimento adicional para validar as soluções propostas pelo mecanismo de adaptação pode ser necessária.

Deve-se ressaltar que as abordagens de adaptação investigadas são limitadas à estratégia de adaptação por substituição. Essas abordagens não podem criar ou remover componentes existentes das soluções recuperadas, mas sim alterar os valores dos componentes para alcançar a solução para um novo problema.

Os resultados obtidos são promissores e sugerem que o conjunto de padrões de adaptação extraídos da BC contém uma boa representação das adaptações necessárias sobre os componentes das soluções do problema sendo resolvido.

Os resultados mostram que, para a maioria dos experimentos, o modelo híbrido de RBC – utilizando como mecanismo de adaptação um comitê composto por um algoritmo SVM como *combinador* – é superior aos outros modelos testados. Os resultados mostram também que, em geral, a utilização do conjunto de dados formado por padrões de adaptação considerando três casos similares ao caso prova produz melhores desempenhos. Os resultados mostram ainda que o mecanismo de recuperação de casos utilizado influencia os resultados finais do sistema – os resultados obtidos mostram que a abordagem de recuperação *Kd-Tree* possibilitou, no geral, um melhor desempenho. Portanto, a utilização de mecanismos de recuperação mais eficientes pode melhorar os resultados obtidos.

Uma das dificuldades encontradas para o desenvolvimento dos trabalhos desta Dissertação de Mestrado foi a obtenção de BCs reais com características que possibilitassem a avaliação empírica das abordagens de adaptação de casos investigadas. Muitas BCs tidas como públicas não são mais acessíveis. As poucas BCs públicas existentes pertencem a domínios de problemas interpretativos, ou seja, problemas nos quais o sistema deve realizar a classificação ou interpretação de uma solução com base nos casos existentes na BC. Esses problemas não demandam adaptação dos valores das soluções recuperadas e portanto não são apropriados para as avaliações das abordagens de adaptação de casos. Como consequência, optou-se pela utilização de conjuntos de dados de AM que possuem características similares às características de uma BC (soluções contendo vários atributos ou com uma grande variabilidade de valores possíveis).

### 7.3 Trabalhos Futuros

Durante o desenvolvimento desta Dissertação de Mestrado surgiram algumas sugestões para o refinamento das abordagens investigadas e novas idéias para o desenvolvimento de outras variações. Essas são apresentadas a seguir como sugestões para futuros trabalhos.

Neste trabalho foi investigada a utilização de comitês híbridos composto por três algoritmos de AM exercendo o papel de estimadores. Como trabalho futuro, pode-se propor a investigação de comitês de algoritmos de AM que utilizem um número maior de estimadores.

Neste trabalho, as abordagens híbridas de adaptação de casos investigadas utilizam a suposição de que a BC original é representativa (Smyth, 1998), ou seja, a BC é uma boa representação dos problemas do domínio da aplicação. Essa suposição é feita para garantir que não será necessário realizar um novo treinamento do mecanismo de adaptação quando novos casos forem integrados à BC. Uma sugestão de trabalho futuro seria integração desse mecanismo de adaptação de casos com políticas de manutenção de conhecimento (Smyth e Keane, 1995; Smyth, 1998; Zhu e Yang, 1999; Leake e Wilson, 2000) para monitorar a evolução da BC do sistema e acionar o processo de extração de padrões de adaptação e treinamento dos algoritmos de AM quando necessário.

As abordagens de adaptação de casos investigadas neste trabalho constituem abordagens de adaptação por substituição. Elas não podem criar novos componentes ou remover componentes existentes nas soluções dos casos. Aqui, pode-se sugerir a integração dessa abordagem de comitês de algoritmos de AM com Algoritmos Evolutivos (AE). Essa nova abordagem alcançaria a estratégia de adaptação por transformação, utilizando o AE para incluir ou remover novos componentes da solução, e a estratégia de adaptação por substituição, utilizando o comitê já investigado para alterar os valores dos componentes existentes e dos novos componentes inseridos pelo AE.

A utilização de métodos de indução lidam com o problema da aquisição de regras para a adaptação de casos. Porém, esses métodos podem não ser capazes de reconhecer possíveis exceções sobre os valores a serem adaptados. Uma possível solução para esse problema, e uma sugestão para um trabalho futuro, é a integração da abordagem investigada neste trabalho com a utilização de casos que constituem exceções aos padrões de adaptação. Esses casos podem ser utilizados quando a entrada do padrão de adaptação é suficientemente similar a um caso exceção recuperado.

Também pode-se sugerir como trabalho futuro a definição e implementação de métricas que permitam medir o esforço necessário para realizar a adaptação de uma solução recuperada. Essas métricas poderiam ser utilizadas como

medida de similaridade durante a recuperação de casos, possibilitando que o mecanismo recuperação identifique os casos que necessitam de um menor esforço de adaptação, tornando o sistema mais eficiente.

## 7.4 Publicações

Nesta Seção são apresentadas as publicações originadas desta Dissertação de Mestrado.

- POLICASTRO, Claudio A., CARVALHO, André C.P.F.L., DELBEM, Alexandre C.B.: Abordagens Híbridas para Recuperação e Adaptação de Casos em Sistemas de Raciocínio Baseado em Casos. NOS Proceedings (CD Media) do XIII Congresso da Sociedade Brasileira de Computação e IV Encontro Nacional de Inteligência Artificial, Campinas, Brasil. (2003)
- POLICASTRO, Claudio A.; CARVALHO, André C.P.L.F.; DELBEM, Alexandre C. B.: Hybrid Approaches for Cases Retrieval and Adaptation. In Proceedings of 26th German Conference on Artificial Intelligence, Hamburg, German. A. Günter, R. Kruse, B. Neumann (Eds.). Lecture Notes in Artificial Intelligence 2821. Springer Verlag. (2003) p. 297-311.
- POLICASTRO, Claudio A.; CARVALHO, André C.P.L.F.; DELBEM, Alexandre C. B.: Hybrid Approaches for Case Adaptation. In Proceedings of 3rd International Conference on Hybrid Intelligent Systems, Melbourne, Austrália. A. Abraham, M. Köppen, K. Franke (Eds.). IOS Press. (2003) p. 331-340.
- POLICASTRO, Claudio A.; CARVALHO, André C.P.L.F.; DELBEM, Alexandre C. B.: Automatic Case Adaptation with a Hybrid Committee Approach. In Proceedings of 3rd Mexican International Conference on Artificial Intelligence, Mexico City, Mexico. R. Monroy, G. Arroyo, L.E. Sucar, H. Sossa (Eds.). Lecture Notes in Artificial Intelligence. Springer Verlag. A ser publicado. (2004)
- POLICASTRO, Claudio A.; CARVALHO, André C.P.L.F.; DELBEM, Alexandre C. B.: A Hybrid Case Based Reasoning Approach for Monitoring Water Quality. In Proceedings of 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Ottawa, Canada. Lecture Notes in Artificial Intelligence. Springer Verlag. A ser publicado. (2004)
- POLICASTRO, Claudio A.; CARVALHO, André C.P.L.F.; DELBEM, Alexandre C. B.: Abordagem Híbrida para Adaptação de Casos em Sistemas de Raciocínio Baseado em Casos. Scientia **14**(2). A ser publicado. (2004)



## Referências

---

- Aamodt, A. e Plaza, E. (1994). Case based reasoning: Foundational issues, methodological variations, and systems approaches. *AI Communications*, 7:39-59.
- Alexandre, F., Labbi, A., Lallement, Y., e Malek, M. (1996). A common architecture for integrating case-based reasoning and neural networks. disponível em <http://citeseer.nj.nec.com/alexandre96common.html>. Relatório Técnico S3, IMAG & INRIA.
- Araujo, D., Lopes, H., e Freitas, A. (2000). Rule discovery with a parallel genetic algorithm. Em *Genetic and Evolutionary Computation Conf Workshop Program, Las Vegas, EUA*, páginas 89-92.
- Batista, G. (2003). *Pré-processamento de Dados em Aprendizado de Máquina Supervisionado*. Tese de Doutorado, Universidade de São Paulo. Disponível em <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-06102003-160219/publico/TeseDoutorado.pdf>.
- Becker, L. e Jazayeri, K. (1989a). A connectionist approach to case-based reasoning. Em Hammond, K., editor, *DARPA Case-Based Reasoning Workshop, Pensacola Beach, Florida*, páginas 213-217. Morgan Kaufmann.
- Becker, L. e Jazayeri, K. (1989b). A neural network model for case-based reasoning. Em Hammond, K., editor, *DARPA Case-Based Reasoning Workshop, Pensacola Beach, Florida*, páginas 334-337. Morgan Kaufmann.
- Bentley, J. (1975). Multidimensional binary search tree used for associative searching. *Machine Learning*, 18(9):509-517.
- Bergmann, R. e Stahl, A. (1998). Similarity measures for object-oriented case representation. Em Smyth, B. e Cunningham, P., editores, *4th. European Workshop on Case-Based Reasoning, Dublin, Ireland*, páginas 25-36. Springer Verlag.

- Blake, C. e Merz, C. (1998). UCI repository of machine learning databases. <http://www.ics.uci.edu/~mllearn/mlrepository.html>. university of california, irvine, dept. of information and computer sciences.
- Braga, A., Ludermir, T., e Carvalho, A. (2000). Sistemas neurais híbridos. Em Braga, A., Carvalho, A., e Ludermir, T., editores, *Redes Neurais Artificiais: Teoria e aplicacoes. Capítulo 7*. Livros Tecnicos e Científicos.
- Carbonel, J. (1986). Derivational analogy: A theory of reconstructive problem solving and expert acquisition. Em Kandel, A. e Langholz, G., editores, *Machine Learning: An AI Approach. Capítulo 14*. Morgan Kaufmann.
- Carpenter, G. e Grossberg, S. (1987). ART 2 : Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26:23:4919–4930.
- Carvalho, D. e Freitas, A. (2000). A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in data. Em *Genetic and Evolutionary Computation Conf, Las Vegas, EUA*, páginas 1061–1068. Morgan Kaufmann.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., e Kegelmeyer, W. P. (2002). SMOTE : Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.
- Corchado, J., Lees, B., e Aiken, J. (2001). Hybrid instance-based system for predicting oceans temperature. *International Journal of Computational Intelligence and Applications*, 1:35–52.
- Corchado, J., Lees, B., Fyle, C., Ress, N., e Aiken, J. (1998). Neuro-adaptation method for a case-based reasoning system. *Computing and Information Systems Journal*, 5:15–20.
- Craw, S. (2003). Introspective learning to build case-based reasoning (CBR) knowledge containers. Em Perner, P. e Rosenfeld, A., editores, *3rd International Conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany*, páginas 1–6. Springer Verlag.
- Craw, S., Jarmulak, J., e Rowe, R. (2001). Using case based data to learn adaptation knowledge for design. Em Nebel, B., editor, *17th. International Joint Conference on Artificial Intelligence, Seattle, EUA*, páginas 1011–1020. Morgan Kaufmann.
- Cunningham, P. e Bonzano, A. (1999). Knowledge engineering in a real world case-based reasoning application. Relatório Técnico 36, Trinity College Dublin. Disponível em [www.cs.tcd.ie/publications/tech-reports/reports.99/TCD-CS-1999-36.pdf](http://www.cs.tcd.ie/publications/tech-reports/reports.99/TCD-CS-1999-36.pdf).

- Domingos, P. (1996). Unifying instance-based and rule-based induction. *Machine Learning*, 24:141–168.
- Duda, R. e Stork, D. (2001). *Pattern Classification*. Wiley-Interscience.
- Falkman, G. (2002). Adaptation using interactive estimations. Em Craw, S. e Preece, A., editores, *6th. European Conference on Case-Based Reasoning, Aberdeen, Scotland, Uk*, páginas 88–102. Springer Verlag.
- Freund, Y. e Schapire, R. (1996). Experiments with a new boosting algorithm. Em Saitta, L., editor, *13th. International Conference on Machine Learning, Bari, Italy*, páginas 148–156. Morgan Kaufmann.
- Garza, A. e Maher, M. (2000). A process model for evolutionary design case adaptation. Em Gero, J., editor, *Artificial Intelligence in Design 00*, páginas 393–412. Kluwer Academic Publishers.
- Gómez, A., Maher, M., e Garza, A. (1999). An evolutionary approach to case adaptation. Em Althoff, K., Bergmann, R., e Branting, K., editores, *3rd. International Conference on Case-Based Reasoning, Munich, Germany*, páginas 162–172. Springer Verlag.
- Hammond, K. (1986). Chef: A model of case based planning. Em *5th National Conference on Artificial Intelligence, Philadelphia, EUA, August*, páginas 267–271. Morgan Kaufmann.
- Hanney, K. (1996). Learning adaptation rules from cases. Tese de Mestrado, University College Dublin. Disponível em <http://www.cs.ucd.ie/students/khanney/MSc.tar.gz>.
- Hanney, K. e Keane, M. (1997). The adaptation knowledge bottleneck: How to ease it by learning from cases. Em Leake, D. e Plaza, E., editores, *2nd. International Conference on Case-Based Reasoning, Providence, Rhode Island, USA*, páginas 359–370. Springer Verlag.
- Haykin, S. (1999). Support vector machines. Em *Neural Networks: A Comprehensive Foundation. Capítulo 6*. Prentice Hall.
- Hilario, M. (1997). An overview of strategies for neurosymbolic integration. Em Sun, R. e Alexandre, F., editores, *Connectionist-Symbolic Processing. Capítulo 2*. Lawrence Earlbaum Associates, Inc.
- Hinkley, D. e Toomey, C. (1994). Clavier: Applying case based reasoning to composite part fabrication. *Innovative Applications of Artificial Intelligence*, 6:54–62.
- Jarmulak, J., Craw, S., e Rowe, R. (2001). Using case-base data to learn adaptation knowledge for design. Em Nebel, B., editor, *17th. International Joint Conference on Artificial Intelligence, Seattle, EUA*, páginas 1011–1020. Morgan Kaufmann.

- Jha, G., Hui, S., e Foo, S. (1999). A hybrid case based reasoning and neural network approach to online intelligent fault diagnosis. Em Parenti, R. e Masulli, F., editores, *3rd. International ICSC Symposia on Intelligent Industrial Automation and Soft Computing, Genova, Italy*, páginas 376–381. ICSC Academic Press.
- Kaufman, K. e Michalski, R. (2000). Ished1: Applying the lem methodology to heat exchanger design. Relatório Técnico 00-2, George Mason University. Disponível em <http://www.mli.gmu.edu/papers/mli00-2.pdf>.
- Kolodner, J. (1992). An introduction to case based reasoning. *AI Review*, 6:3–34.
- Kolodner, J. (1993). Adaptation methods and strategies. Em Kolodner, J., editor, *Case-Based Reasoning. Capítulo 11*. Morgan Kaufmann.
- Krovvidy, S. e Wee, W. (1992). An intelligent hybrid system for wastewater treatment. Em Kandel, A. e Langholz, G., editores, *Hybrid Architecture for Intelligent Systems. Capítulo 17*. CRC Press.
- Leake, D. (1995). Becoming an expert case-based reasoner: learning to adapt prior cases. Em *8th. Annual Florida Artificial Intelligence Research Symposium, Melbourne*, páginas 112–116.
- Leake, D. (1996). CBR in context: The present and future. Em Leake, D., editor, *Case-Based Reasoning: Experiences, Lessons and Future Directions. Capítulo 1*. AAAI Press/MIT Press.
- Leake, D., Kinley, A., e Wilson, D. (1996). Acquiring case adaptation knowledge: A hybrid approach. Em Burkhard, D. e Lenz, M., editores, *30th. National Conference on Artificial Intelligence and 8th. Innovative Applications of Artificial Intelligence Conference, Portland, USA*, páginas 684–689. AAAI Press/MIT Press.
- Leake, D. e Wilson, D. (1998). Categorizing case-base maintenance: Dimensions and directions. Em Smyth, B. e Cunningham, P., editores, *4th. European Workshop on Case-Based Reasoning, Dublin, Ireland*, páginas 196–207. Springer Verlag.
- Leake, D. e Wilson, D. (2000). Remembering why to remember: Performance-guided case-base maintenance. Em Blanzieri, E. e Portinale, L., editores, *5th. European Workshop on Case-Based Reasoning, Trento, Italy*, páginas 161–172. Springer Verlag.
- Lenz, M., B. H.-D. (1996a). Case retrieval nets: Basic ideas and extensions. Em Burkhard, H.-D. e Lenz, M., editores, *4th German Workshop on Case-Based Reasoning: System Development and Evaluation, Berlin, German*, páginas 103–110.

- Lenz, M. (1996b). Case retrieval nets: Basic ideas and extensions. Em Gorz, G. e Holldobler, S., editores, *20th German Conference on Artificial Intelligence, Dresden, German, September 1996*, páginas 227-239. Springer Verlag.
- Lotfy, E. e Sameh, A. (2002). Applying neural networks in case based reasoning adaptation for cost assessment of steel buildings. *International Journal of Computers and Applications*, 24:28-38.
- Main, J., Dillom, T., e Shiu, S. (2001). A tutorial on case based reasoning. Em Pal, S., Dillon, T., e Yeung, D., editores, *Soft Computing in Case Based Reasoning. Capítulo 1*. Springer Verlag.
- Main, J. e Dillon, T. (1999). A hybrid case-based reasoner for footwear design. Em Althoff, K., Bergmann, R., e Branting, K., editores, *3rd. International Conference on Case-Based Reasoning, Munich, Germany*, páginas 497-509. Springer Verlag.
- Main, J. e Dillon, T. (2001). A neuro-fuzzy methodology for case retrieval and an object-oriented case schema for structuring case bases and their application to fashion footwear design. Em Pal, S., Dillon, T., e Yeung, D., editores, *Soft Computing in Case Based Reasoning. Capítulo 12*. Springer Verlag.
- Malek, M. (2001). Hybrid approaches for integrating neural networks and case-based reasoning: From loosely coupled to tightly coupled models. Em Pal, S., Dillon, T., e Yeung, D., editores, *Soft Computing in Case Based Reasoning. Capítulo 4*. Springer Verlag.
- Malek, M. e Amy, B. (1994). Integration of case-based reasoning and neural networks approaches for classification. Relatório Técnico 131 IMAG - 28 LIFIA, laboratoire Leibntz - IMAG. Disponível em <http://www-leibniz.imag.fr/RESEAUX/pub/malek.cbrnn.e.ps.gz>.
- Michalski, R., Carbonell, J., e Mitchell, T. (1983). *Machine Learning: An Artificial Intelligence Approach*. Morgan Kaufmann.
- Mitchell, T. (1997a). Evaluating hypotheses. Em *Machine Learning. Capítulo 5*. McGraw-Hill.
- Mitchell, T. (1997b). Instance-based learning. Em *Machine Learning. Capítulo 8*. McGraw-Hill.
- Monard, M. e Baranauskas, J. (2003). Conceitos sobre aprendizado de máquina. Em Rezende, S., editor, *Sistemas Inteligentes: Fundamentos e Aplicações. Capítulo 4*. Manole.

- Moses, L. (1986). Comparison of averages from two samples and some related problems. Em *Think and Explain with Statistics. Capítulo 6*. Addison – Wesley.
- Myllimaki, P. e Tirri, H. (1994). Massively parallel case-based reasoning with probabilistic similarity metrics. Em Wess, S., Althoff, K., e Richter, M., editores, *1st European Workshop on Case-Based Reasoning*, páginas 144–154. Springer Verlag.
- Orr, M. (1996). Introduction to radial basis function networks. Relatório técnico, Centre for Cognitive Science. University of Edinburgh. Disponível em <http://www.anc.ed.ac.uk/mjo/papers/intro.ps>.
- Plaza, E. e Arcos, J. (2002). Constructive adaptation. Em Craw, S. e Preece, A., editores, *6th. European Conference on Case-Based Reasoning, Aberdeen, Scotland, Uk*, páginas 306–320.
- Policastro, C., Carvalho, A., e Belbem, A. (2003a). Automatic case adaptation with a hybrid committee approach. Em Abraham, A., Köppen, M., e Franke, K., editores, *3rd international Conference on Hybrid Intelligent Systems, Melbourne, Austrália*, páginas 331–340. IOS Press.
- Policastro, C., Carvalho, A., e Belbem, A. (2003b). Hybrid approaches for cases retrieval and adaptation. Em Günter, A., Kruse, R., e Neumann, B., editores, *26th German Conference on Artificial Intelligence, Hamburg, German*, páginas 297–311. Springer Verlag.
- Policastro, C., Carvalho, A., e Belbem, A. (2003c). Hybrid approaches for cases retrieval and adaptation. Em Monroy, R., Arroyo, G., Sucar, L., e Sossa, H., editores, *3rd Mexican International Conference on Artificial Intelligence, Mexico City, Mexico. A ser publicado*. Springer Verlag.
- Prentzas, J. e Hatzilygeroudis, I. (2002). Integrating hybrid rule-based with case-based reasoning. Em Craw, S. e Preece, A., editores, *6th. European Conference on Case-Based Reasoning, Aberdeen, Scotland, Uk*, páginas 336–349. Springer Verlag.
- Pérez, E., Coello, C., e Aguirre, A. (2002). Genetic algorithms and case-based reasoning as a discovery and learning machine in the optimization of combinational logic circuits. Em Coello, C., Albornoz, A., Sucar, L., e Battistutti, O., editores, *2nd. Mexican International Conference on Artificial Intelligence, Merida, Mexico*, páginas 128–137. Springer Verlag.
- Puvis, L. e Athalye, S. (1997). Towards improving case adaptability with a genetic algorithm. Em Leake, D. e Plaza, E., editores, *2nd. International Conference on Case-Based Reasoning, Providence, Rhode Island, USA*, páginas 403–412. Springer Verlag.

- Quinlan, J. (1992). Learning with continuous classes. Em *5th. Australian Joint Conference on Artificial Intelligence, Hobart, Tasmania, World Scientific, Singapore*, páginas 343–348.
- Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Racine, K. e Yang, Q. (1997). Maintaining unstructured case bases. Em Leake, D. e Plaza, E., editores, *2nd. International Conference on Case-Based Reasoning, Providence, Rhode Island, USA*, páginas 553–564. Springer Verlag.
- Reategui, E. (1995). Using a neural network to learn general knowledge in a case-based system. Em Veloso, M. e Aamodt, A., editores, *1st. International Conference on Case-Based Reasoning, Sesimbra, Portugal*, páginas 528–537. Springer Verlag.
- Reategui, E. e Campbell, J. (1994). A classification system for credit card transaction. Em M.Keane, editor, *2th. European Workshop on Case-Based Reasoning, Chantilly, France*, páginas 280–291. Springer Verlag.
- Reinartz, T., Iglezakis, I., e Berghofer, T. (2000). On quality measures for case base maintenance. Em E.Blanzieri e L.Portinale, editores, *5th. European Workshop on Case-Based Reasoning, Trento, Italy*, páginas 247–259. Springer Verlag.
- Richter, M. (1995). The knowledge contained in similarity measures. Invited Talker at 1st. International Conference on Case-Based Reasoning, Sesimbra, Portugal. Disponível em <http://www.cbr-web.org/documents/richterccbr95remarks.html>.
- Schaaf, J. (1996). Fish and shrink. a next step towards efficient case retrieval in large case bases. Em Smith, I. e Faltings, B., editores, *3rd European Workshop on Case-Based Reasoning, Lausanne, Switzerland, November 1996*, páginas 362–376. Springer Verlag.
- Shiu, S., Wang, X., e Yeung, D. (2001). Neuro-fuzzy approach for maintaining case bases. Em Pal, S., Dillon, T., e Yeung, D., editores, *Soft Computing in Case Based Reasoning. Capítulo 11*. Springer Verlag.
- Smyth, B. (1998). Case base maintenance. Em Mira, J. e Pobil, A., editores, *12th. International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Cairo, Egypt*, páginas 507–516. Springer Verlag.
- Smyth, B. e Cunningham, P. (1993). Complexity of adaptation in real-world case-based reasoning systems. Em *6th. Irish Conference on Artificial Intelligence and Cognitive Science, Belfast*, páginas 228–240.

- Smyth, B. e Keane, M. (1995). Remembering to forget: A competence preserving case deletion policy for CBR systems. Em Weiß, G. e Sen, S., editores, *14th. International Joint Conference on Artificial Intelligence, Montréal, Canada*, páginas 377–382. Morgan Kaufmann.
- Smyth, B. e Keane, M. (1998). Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence*, 102:249–293.
- Smyth, B. e McKenna, E. (1999). Footprint-based retrieval. Em Althoff, K., Bergmann, R., e Branting, K., editores, *3rd International Conference on Case-Based Reasoning, Munich, Germany*, páginas 343–357. Springer Verlag.
- Sovat, R. e Carvalho, A. (2001). Retrieval and adaptation of cases using a neural network. Em Wahlster, W., editor, *4th. International Conference on Case-Based Reasoning, Vancouver, Canada*, páginas 196–200. Springer Verlag.
- Sporl, B., Lenz, M., e Hubner, A. (1999). Case-based reasoning - survey and future directions. Em *5th. Biannual German Conference on Knowledge Based Systems, Würzburg, Germany*, páginas 67–89. Springer Verlag.
- Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons.
- Varma, A. (1999). Icarus: Design and deployment of a case-based reasoning system for locomotive diagnostics. Em Althoff, K., Bergmann, R., e Branting, K., editores, *3rd. International Conference on Case-Based Reasoning, Munich, Germany*, páginas 581–595. Springer Verlag.
- Wang, Y. e Witten, I. (1997). Induction of model trees for predicting continuous classes. Em Someren, M. e Widmer, G., editores, *9th. European Conference on Machine Learning, Prague, Czech Republic*, páginas 128–137. Springer Verlag.
- Wangenheim, C. (2000). Case-based reasoning - a short introduction. Relatório Técnico 02/00, Federal University of Santa Catarina. Disponível em [http://www.inf.ufsc.br/~awangenh/RP/RBC/intro\\_rbc.pdf](http://www.inf.ufsc.br/~awangenh/RP/RBC/intro_rbc.pdf).
- Watson, I. (1999). CBR is a methodology not a technology. *Knowledge-Based Systems*, 12:303–308.
- Watson, I. (2002). Applying knowledge management: Techniques for building organizational memories. Em Craw, S. e Preece, A., editores, *6th. European Conference on Case-Based Reasoning, Aberdeen, Scotland, UK*, páginas 6–12. Springer Verlag.
- Watson, I. e Marir, F. (1994). Case based reasoning: A review. *The Knowledge Engineering Review*, 9:355–381.



- Wiratunga, N., Craw, S., e Rowe, R. (2002). Learning to adapt for case-based design. Em Craw, S. e Preece, A., editores, *6th. European Conference on Case-Based Reasoning, Aberdeen, Scotland, Uk*, páginas 421–435. Springer Verlag.
- Zhu, J. e Yang, Q. (1999). Remembering to add: Competence-preserving case-addition policies for case-base maintenance. Em Dean, T., editor, *16th. International Joint Conference on Artificial Intelligence, Stockholm, Sweden*, páginas 234–239. Morgan Kaufmann.