
Ajuste de parâmetros de técnicas de classificação por
algoritmos bioinspirados

André Luis Debiaso Rossi

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 20 de fevereiro de 2009

Assinatura: _____

Ajuste de parâmetros de técnicas de classificação por algoritmos bioinspirados

André Luis Debiaso Rossi

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 de Computação e Matemática Computacional.

USP – São Carlos
Fevereiro/2009

Dedicatória

Aos meus Pais, João e Maria.

Agradecimentos

Em primeiro lugar gostaria de agradecer a Deus, por estar sempre tão presente em minha vida.

Ao meu orientador Prof. Dr. André Carvalho, pela orientação durante os dois anos de mestrado, pela ajuda nesta dissertação, por proporcionar as condições necessárias para o desenvolvimento deste trabalho, pela confiança depositada em mim, pelos momentos de positivismo e pela amizade.

Ao Prof. Dr. Carlos Soares do LIAAD, Porto - Portugal, pela contribuição com este trabalho, pela recepção em Porto e pelo seu entusiasmo como pesquisador.

Aos meus pais João e Maria Inês, pelo amor incondicional e pelo exemplo de honestidade e humildade.

Aos meus irmãos Flávio, Regina e Ana Elisa, pelas brincadeiras e por estarem sempre prontos para ajudar.

À minha namorada Joyce, pelo apoio, demonstrações de amor e pelo seu sorriso, que me faz tão bem.

Aos meus familiares, que me apoiam desde a graduação.

Aos amigos e professores do BIOCAMP e do LABIC, pelo acolhimento, pelos ensinamentos e também pelos momentos de descontração. Em especial à Ronaldo Cristiano Prati, pelos algoritmos dos testes estatísticos e pela ajuda com o \LaTeX , à Ana Carolina Lorena, pela ajuda e correções do projeto que se tornaria este trabalho e à Bruno Feres de Souza, pelos conjuntos de dados e ensinamentos.

Aos amigos com quem morei, mesmo que por pouco tempo, durante esses dois anos em São Carlos: Fábio, Vinícius e Régis, pelo incentivo e companheirismo.

A todos os funcionários do ICMC da USP, pela competência e dedicação.

À FAPESP e ao CNPq, pelo apoio financeiro para a realização deste estudo.

Resumo

Aprendizado de máquina é uma área de pesquisa na qual se investiga como desenvolver sistemas capazes de aprender com a experiência. Muitos algoritmos de aprendizado possuem parâmetros cujos valores devem ser especificados pelo usuário. Em geral, esses valores influenciam diretamente no processo de aquisição do conhecimento, podendo gerar diferentes modelos. Recentemente, algoritmos de otimização bioinspirados têm sido aplicados com sucesso no ajuste de parâmetros de técnicas de aprendizado de máquina. Essas técnicas podem apresentar diferentes sensibilidades em relação aos valores escolhidos para seus parâmetros e diferentes algoritmos de ajuste de parâmetros podem apresentar desempenhos singulares. Esta dissertação investiga a utilização de algoritmos bioinspirados para o ajuste de parâmetros de redes neurais artificiais e máquinas de vetores de suporte em problemas de classificação. O objetivo dessa investigação é verificar quais são as técnicas que mais se beneficiam do ajuste de parâmetros e quais são os algoritmos mais eficientes para essas técnicas. Os resultados experimentais mostram que os algoritmos bioinspirados conseguem encontrar melhores classificadores que outras abordagens. Porém, essa melhoria é estatisticamente significativa apenas para alguns conjuntos de dados. Foi possível verificar que o uso dos valores padrão para os parâmetros das técnicas de classificação leva a desempenhos similares aos obtidos com os algoritmos bioinspirados. Entretanto, para alguns conjuntos de dados, o ajuste de parâmetros pode melhorar significativamente o desempenho de um classificador.

Abstract

Machine learning is a research area whose main goal is to design computational systems capable of learning through experience. Many machine learning techniques have free parameters whose values are generally defined by the user. Usually, these values directly affect the knowledge acquisition process directly, resulting in different models. Recently, bioinspired optimization algorithms have been successfully applied to the parameter tuning of machine learning techniques. These techniques may present variable sensitivity to the selection of the values of its parameters and different parameter tuning algorithms may present different behaviors. This thesis investigates the use of bioinspired algorithms for the parameter tuning of artificial neural networks and support vector machines in classification problems. The goal of this thesis is to investigate which techniques benefits most from parameter tuning and which are the most efficient algorithms to use with these techniques. Experimental results show that these bioinspired algorithms can find better classifiers when compared to other approaches. However, this improvement is statistically significant only to some datasets. It was possible to verify that the use of standard parameter values for the classification techniques leads to similar performances to those obtained with the bioinspired algorithms. However, for some datasets, the parameter tuning may significantly improve a classifier performance.

Esta dissertação foi preparada com o formatador de textos L^AT_EX. A bibliografia é gerada automaticamente pelo B_IB_TE_X, utilizando o estilo *apalike* com modificações para o português. O estilo utilizado no documento e as modificações no estilo *apalike* foram desenvolvidos por Ronaldo Cristiano Prati.

©Copyright 2009 – André Luis Debiaso Rossi
Todos os Direitos Reservados

Sumário

Dedicatória	v
Agradecimentos	vii
Resumo	ix
Abstract	xi
Sumário	xv
Lista de Abreviaturas	xvii
Lista de Figuras	xix
Lista de Tabelas	xxi
Lista de Algoritmos	xxv
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	3
1.3 Métodos e Técnicas Investigadas	3
1.4 Principais Resultados Experimentais	4
1.5 Organização do Texto	5
2 Aprendizado de Máquina	7
2.1 Classificação	8
2.1.1 Problemas de Classificação em Bioinformática	9
2.1.2 Avaliação de Classificadores	10
2.2 Máquinas de Vetores de Suporte	14
2.3 Redes Neurais Artificiais	16
2.4 Seleção de Modelo e Ajuste de Parâmetros	17
2.5 Considerações Finais	21

3	Computação Natural	23
3.1	Inteligência de Enxames	24
3.1.1	Otimização por Colônia de Formigas	25
3.1.2	Otimização por Enxame de Partículas	26
3.2	Sistema Imunológico Artificial	29
3.3	Algoritmos Genéticos	31
3.4	Considerações Finais	34
4	Experimentos com Funções de <i>Benchmark</i>	35
4.1	Funções de <i>Benchmark</i>	36
4.2	Algoritmos Bioinspirados Implementados	38
4.3	Resultados	40
4.4	Considerações Finais	42
5	Experimentos de Ajuste de Parâmetros	45
5.1	Conjuntos de Dados	46
5.2	Algoritmos Bioinspirados	48
5.3	Técnicas de Classificação Investigadas	49
5.4	Avaliação dos Experimentos	50
5.5	Método Experimental A: Ajuste de Parâmetros de SVMs	51
5.6	Método Experimental B: Ajuste de Parâmetros de RNs	55
5.7	Método Experimental C: Ajuste de Parâmetros de RNs e SVMs	64
5.7.1	Resultados do Ajuste de Parâmetros de RNs com o Método Experimental C	66
5.7.2	Resultados do Ajuste de Parâmetros de SVMs com o Método Experimental C	73
5.7.3	Comparação entre RNs e SVMs no Domínio de Expressão Gênica	81
5.8	Considerações Finais	83
6	Conclusão	85
6.1	Principais Resultados	86
6.2	Contribuições	88
6.3	Limitações, Problemas Encontrados e Trabalhos Futuros	89
	Referências Bibliográficas	91
A	Densidade das Taxas de Erro de Validação e de Teste	105
B	Curvas ROC	121

Lista de Abreviaturas

ACO Otimização por Colônia de Formigas (do inglês, *Ant Colony Optimization*)

AD Árvore de Decisão

AG Algoritmo Genético

Ag Antígeno (do inglês, *Antigen*)

AIC *Akaike Information Criterion*

AM Aprendizado de Máquina

AUC Área sob uma Curva ROC (do inglês, *Area Under an ROC Curve*)

BSS *Between-groups Sum of Squares*

CE Computação Evolutiva

CSA Algoritmo de Seleção Clonal (do inglês, *Clonal Selection Algorithm*)

CV Validação Cruzada (do inglês, *Cross-validation*)

DNA Ácido Desoxirribonucléico (do inglês, *Desoxyribonucleic Acid*)

FDP Função de Densidade de Probabilidade

FN Falso Negativo

FP Falso Positivo

HTGA *hybrid Taguchi-Genetic Algorithm*

IA Algoritmo Imunológico (do inglês, *Immune Algorithm*)

IC Inteligência Computacional

McESE *MacMaster Expert System Environment*

MLP Perceptron multi-camadas (do inglês, *Multi-Layer Perceptron*)

mRNA Ácido Ribonucléico mensageiro (do inglês, *Messenger Ribonucleic Acid*)

PDB *Protein Data Bank*

PIR *Protein Information Resource*

PSO Otimização por Enxame de Partículas (do inglês, *Particle Swarm Optimization*)

RBF Função de Base Radial (do inglês, *Radial Basis Function*)

RF *Random Forests*

RN Rede Neural Artificial

RNA Ácido Ribonucléico (do inglês, *Ribonucleic Acid*)

ROC *Receiver Operating Characteristics*

SAGE *Serial Analysis of Genes Expression*

SCOP *Structural Classification of Protein*

SI Sistema Imunológico

SIA Sistema Imunológico Artificial

SVM Máquina de Vetores de Suporte (do inglês, *Support Vector Machine*)

TFP Taxa de Falsos Positivos

TNoM *Threshold Number of Misclassification*

TVP Taxa de Verdadeiros Positivos

UCI *University of California - Irvine*

VN Verdadeiro Negativo

VP Verdadeiro Positivo

WEKA *Waikato Environment for Knowledge Analysis*

WSS *Within-groups Sum of Squares*

Lista de Figuras

2.1	Diagrama do processo de indução de um classificador e sua utilização na dedução de novos exemplos.	9
2.2	Exemplo de curva ROC.	13
2.3	A idéia do hiperplano ótimo para exemplos linearmente separáveis.	15
2.4	Neurônio artificial e o esquema de uma rede <i>feed-forward</i>	17
3.1	Formigas em busca de alimento. Ao passar do tempo, o menor caminho possuirá a maior quantidade de feromônio depositado.	25
3.2	Representação da vizinhança topológica para partículas.	28
3.3	Representação da vizinhança global (à esquerda) e local (à direita).	28
3.4	Representação de um cromossomo binário em um AG.	31
3.5	Operador genético de <i>crossover</i> de um ponto (Souto et al., 2003).	32
3.6	Operador genético de mutação (Souto et al., 2003).	33
4.1	Gráfico das funções de <i>benchmark</i> utilizadas.	37
4.2	Convergência dos algoritmos bioinspirados para funções unimodais.	41
4.3	Convergência dos algoritmos bioinspirados para funções multimodais.	43
5.1	Método Experimental A.	51
5.2	Método Experimental B.	56
5.3	Representação de um indivíduo para ajuste de parâmetros de RNs.	57
A.1	Densidade da taxa de erro de validação e de teste para o conjunto de dados Cólón para todas as combinações de valores de parâmetros para RNs.	106
A.2	Densidade da taxa de erro de validação e de teste para o conjunto de dados Glioma para todas as combinações de valores de parâmetros para RNs.	107
A.3	Densidade da taxa de erro de validação e de teste para o conjunto de dados Leucemia para todas as combinações de valores de parâmetros para RNs.	108
A.4	Densidade da taxa de erro de validação e de teste para o conjunto de dados Pâncreas para todas as combinações de valores de parâmetros para RNs.	109
A.5	Densidade da taxa de erro de validação e de teste para o conjunto de dados Cólón para todas as combinações de valores de parâmetros para SVMs.	110

A.6	Densidade da taxa de erro de validação e de teste para o conjunto de dados Glioma para todas as combinações de valores de parâmetros para SVMs.	111
A.7	Densidade da taxa de erro de validação e de teste para o conjunto de dados Leucemia para todas as combinações de valores de parâmetros para SVMs.	112
A.8	Densidade da taxa de erro de validação e de teste para o conjunto de dados Pâncreas para todas as combinações de valores de parâmetros para SVMs.	113
A.9	Densidade da taxa de erro de validação e de teste para o conjunto de dados Leucemia2 para todas as combinações de valores de parâmetros para SVMs.	114
A.10	Densidade da taxa de erro de validação e de teste para o conjunto de dados Pulmão para todas as combinações de valores de parâmetros para SVMs.	115
A.11	Densidade da taxa de erro de validação e de teste para o conjunto de dados <i>australian</i> para todas as combinações de valores de parâmetros para SVMs.	116
A.12	Densidade da taxa de erro de validação e de teste para o conjunto de dados <i>pima</i> para todas as combinações de valores de parâmetros para SVMs.	117
A.13	Densidade da taxa de erro de validação e de teste para o conjunto de dados <i>segment</i> para todas as combinações de valores de parâmetros para SVMs.	118
A.14	Densidade da taxa de erro de validação e de teste para o conjunto de dados <i>vehicle</i> para todas as combinações de valores de parâmetros para SVMs.	119
B.1	Curvas ROC para o conjunto de dados Cólón usando RNs com parâmetros ajustados.	121
B.2	Curvas ROC para o conjunto de dados Glioma usando RNs com parâmetros ajustados.	121
B.3	Curvas ROC para o conjunto de dados Leucemia usando RNs com parâmetros ajustados.	122
B.4	Curvas ROC para o conjunto de dados Pâncreas usando RNs com parâmetros ajustados.	122
B.5	Curvas ROC para o conjunto de dados Cólón usando SVMs com parâmetros ajustados.	122
B.6	Curvas ROC para o conjunto de dados Glioma usando SVMs com parâmetros ajustados.	122
B.7	Curvas ROC para o conjunto de dados Leucemia usando SVMs com parâmetros ajustados.	123
B.8	Curvas ROC para o conjunto de dados Pâncreas usando SVMs com parâmetros ajustados.	123
B.9	Curvas ROC para o conjunto de dados <i>australian</i> usando SVMs com parâmetros ajustados.	123
B.10	Curvas ROC para o conjunto de dados <i>pima</i> usando SVMs com parâmetros ajustados.	123

Lista de Tabelas

2.1	Conjunto de dados para o diagnóstico da saúde de pacientes.	8
2.2	Matriz de confusão.	11
2.3	Principais <i>kernels</i> utilizados nas SVMs (Haykin, 1999).	16
4.1	Funções de <i>benchmark</i> utilizadas para verificação da funcionalidade e desempenho dos algoritmos bioinspirados investigados.	36
4.2	Valores para os parâmetros dos algoritmos bioinspirados.	39
4.3	Resultados para cada algoritmo bioinspirado para as funções unimodais após 10 000 iterações.	40
4.4	Resultados para cada algoritmo bioinspirado para as funções multimodais após 10 000 iterações.	42
5.1	Conjuntos de dados utilizados nos experimentos.	46
5.2	Valores para os parâmetros dos algoritmos bioinspirados.	49
5.3	Cólon - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Cólon (em %).	52
5.4	Glioma - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Glioma (em %).	53
5.5	Leucemia - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Leucemia (em %).	53
5.6	Pâncreas - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Pâncreas (em %).	54
5.7	Intervalos de valores encontrados para os parâmetros γ e C	54
5.8	Cólon - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Cólon.	59
5.9	Glioma - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Glioma.	60
5.10	Leucemia - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Leucemia.	60

5.11	Pâncreas - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Pâncreas.	60
5.12	Análise estatística - teste- <i>t</i> pareado corrigido (conjunto de dados de validação).	61
5.13	<i>Ranking</i> Médio usando o teste de Friedman.	62
5.14	Estatísticas dos melhores valores de parâmetros encontrados para o conjunto de validação para todos os algoritmos bioinspirados.	63
5.15	Medidas da taxa de erro de cada conjunto de dados (média para as 10 partições).	63
5.16	Cólon - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Cólon.	67
5.17	Glioma - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Glioma.	67
5.18	Leucemia - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Leucemia.	67
5.19	Pâncreas - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Pâncreas.	67
5.20	Cólon - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Cólon.	68
5.21	Glioma - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Glioma.	68
5.22	Leucemia - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Leucemia.	68
5.23	Pâncreas - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Pâncreas.	68
5.24	Correlação das taxas de erro de validação e de teste para todas as combinações de parâmetros.	69
5.25	Análise estatística - teste- <i>t</i> pareado corrigido (conjunto de dados de validação).	70
5.26	<i>Ranking</i> Médio usando o teste de Friedman.	71
5.27	Medidas da taxa de erro de cada conjunto de dados (média para as 10 partições).	72
5.28	Estatísticas dos melhores valores de parâmetros encontrados por todos os algoritmos bioinspirados.	72

5.29	Área abaixo da curva ROC para classificadores gerados utilizando valores dos parâmetros cuja taxa de erro mais se aproxima da média.	73
5.30	Cólon - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Cólon.	74
5.31	Glioma - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Glioma.	74
5.32	Leucemia - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Leucemia.	74
5.33	Pâncreas - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Pâncreas.	74
5.34	Leucemia2 - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Leucemia2.	75
5.35	Pulmão - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Pulmão.	75
5.36	Australian - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Australian.	75
5.37	Pima-indians-diabetes - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Pima-indians-diabetes.	75
5.38	Segment - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Segment.	76
5.39	Vehicle - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Vehicle.	76
5.40	Análise estatística - teste t pareado corrigido (conjunto de dados de teste).	78
5.41	<i>Ranking</i> médio usando o teste de Friedman.	78
5.42	Correlação das taxas de erro de validação e de teste para todas as combinações de parâmetros.	79
5.43	Medidas da taxa de erro de cada conjunto de dados (média para as 10 partições).	80
5.44	Estatísticas dos melhores valores de parâmetros encontrados por todos os algoritmos bioinspirados (2^v , em que v é o valor encontrado para o parâmetro).	80
5.45	Área abaixo da curva ROC para classificadores gerados utilizando valores dos parâmetros cuja taxa de erro mais se aproxima da média.	81

5.46	Comparação entre RNs e SVMs para classificação de dados de expressão gênica.	82
------	--	----

Lista de Algoritmos

1	Algoritmo ACO _{\mathbb{R}}	27
2	Algoritmo PSO.	29
3	Algoritmo CLONALG para problemas de otimização.	31
4	Algoritmo Genético.	33
5	Método experimental C para ajuste de parâmetros.	65

Capítulo 1

Introdução

Neste trabalho foi investigada a utilização de algoritmos computacionais inspirados na natureza, ou bioinspirados, para ajustar os valores dos parâmetros livres de técnicas de Aprendizado de Máquina (AM). Foi realizada uma comparação entre abordagens clássicas e os algoritmos bioinspirados para o ajuste de parâmetros, avaliando o desempenho obtido pelas técnicas de AM na classificação de dados, principalmente do domínio de expressão gênica.

Com o passar dos anos, a computação tem sido utilizada para solucionar problemas das mais diversas áreas do conhecimento. Porém, algoritmos e linguagens de programação clássicas não são capazes de solucionar muitos problemas advindos das novas necessidades de utilização dos computadores. O surgimento da área de Inteligência Computacional (IC), criou inúmeras expectativas. Entretanto, em várias situações, os resultados obtidos ficaram muito aquém dos esperados.

Aprendizado de Máquina é uma área de pesquisa que estuda como desenvolver sistemas capazes de adquirir conhecimento de forma automática. A forma de obtenção desse conhecimento é o que leva pesquisadores dessa área a desenvolverem técnicas computacionais que tornam possível o aprendizado (Monard & Baranauskas, 2003). Técnicas de AM têm alcançado grande sucesso na execução de tarefas específicas, como reconhecimento da fala, mineração de dados, reconhecimento de padrões, além de muitas outras (Mitchell, 1997). Na Bioinformática, as técnicas de AM têm sido amplamente utilizadas em problemas como reconhecimento de genes, análise de expressão gênica e predição da estrutura de proteínas. A análise de dados biológicos desperta grande interesse da comunidade científica por auxiliar a criação de novas drogas e tratamentos para doenças.

Grande parte dos algoritmos de AM possuem parâmetros cujos valores devem ser especificados pelo usuário. Os valores para esses parâmetros livres, em geral, influenciam diretamente no desempenho de modelos induzidos (Kohavi & John, 1995). O problema de encontrar os melhores valores para os parâmetros livres é denominado de ajuste de parâmetros. A utilização de métodos de busca exaustiva pode tornar-se impraticável, devido ao número de parâmetros e ao grande número de possíveis valores para alguns

parâmetros.

Diferentes algoritmos e métodos são utilizados para o ajuste de valores de parâmetros de técnicas de AM, como algoritmos de busca exaustiva e métodos tradicionais de otimização. Atualmente, como uma alternativa a esses métodos, algoritmos bioinspirados são empregados para o ajuste de parâmetros. Esses algoritmos fazem parte de uma área denominada Computação Natural, a qual é formada por três frentes de estudo: a inspiração na natureza para desenvolvimento de novas técnicas computacionais; a simulação e emulação de fenômenos naturais e a utilização de novos materiais para realizar computação (Castro, 2007).

Na Seção 1.1 é apresentada a motivação para a realização deste estudo. Os objetivos são descritos na Seção 1.2. Na Seção 1.3 são apresentados os métodos e as técnicas investigadas. Na Seção 1.4 são mostrados os principais resultados obtidos nos experimentos. Por último, na Seção 1.5, é descrita a organização geral dessa dissertação.

1.1 Motivação

A influência dos valores dos parâmetros livres no desempenho de técnicas de AM têm levado pesquisadores a desenvolverem diferentes algoritmos e técnicas a fim de melhorar o desempenho obtido, reduzir o custo computacional e automatizar o processo de ajuste de parâmetros.

Para RNs, por exemplo, Basheer & Hajmeer (2000) afirmam que bons valores para os parâmetros são encontrados, geralmente, por meio de tentativa e erro. Segundo os autores, a escolha de valores para os parâmetros do algoritmo de aprendizado *backpropagation* para RNs influencia na convergência do aprendizado e no desempenho geral da rede.

Diversos trabalhos que utilizam algoritmos bioinspirados com o objetivo de ajustar os parâmetros de técnicas de AM já foram publicados. Chunhong & Licheng (2004) e Lorena & Carvalho (2006) utilizaram algoritmos genéticos para ajustar os parâmetros de máquinas de vetores de suporte, enquanto que Souza et al. (2006) utilizaram otimização por enxame de partículas para a mesma técnica de AM. Em Lacerda et al. (2002) e Leung et al. (2003) os autores ajustaram os parâmetros de redes neurais utilizando algoritmos genéticos.

Contudo, não é do nosso conhecimento trabalhos comparando resultados entre diferentes técnicas bioinspiradas aplicadas à diferentes técnicas de AM. O único trabalho encontrado na literatura que compara algoritmos genéticos para ajustar parâmetros de dois algoritmos de AM, TIMBL (Daelemans et al., 2002) e RIPPER (Cohen, 1995) foi de Hoste & Daelemans (2005).

Diferentes técnicas de AM podem apresentar diferentes sensibilidades referentes à escolha dos valores de seus parâmetros. Da mesma maneira, diferentes métodos para ajustar esses parâmetros podem apresentar desempenhos distintos. Tendo essas afirmações como

verdadeiras, seria interessante avaliar a utilização de diferentes métodos para ajuste de parâmetros de diferentes técnicas de AM.

1.2 Objetivos

Neste estudo foi investigada a utilização de algoritmos bioinspirados para o ajuste dos valores dos parâmetros livres de técnicas de AM. Este trabalho tem os seguintes objetivos:

- verificar se a otimização dos valores dos parâmetros de diferentes técnicas de AM faz com que essas apresentem desempenhos semelhantes. Como as técnicas possuem diferentes vies indutivos, espera-se que isso não ocorra;
- verificar se os algoritmos de otimização bioinspirados possuem desempenhos semelhantes na otimização dos valores dos parâmetros livres. Pela mesma razão anterior, também espera-se que isso não ocorra;
- comparar os resultados obtidos por algoritmos de otimização bioinspirados com outros métodos para esse problema, como valores padrões para os parâmetros e busca exaustiva;
- verificar quais são as técnicas de AM que mais se beneficiam do ajuste de parâmetros e quais métodos de ajuste se mostram mais eficientes para diferentes técnicas de AM.

1.3 Métodos e Técnicas Investigadas

Neste trabalho, quatro algoritmos bioinspirados são aplicados para ajustar os parâmetros de duas técnicas de AM. Os algoritmos são baseados em: Otimização por Enxame de Partículas (PSO, do inglês, *Particle Swarm Optimization*) (Kennedy & Eberhart, 1995), Algoritmo de Seleção Clonal (CSA, do inglês, *Clonal Selection Algorithm*) (Castro & Von-Zuben, 2002), que faz parte dos Sistemas Imunológicos Artificiais (SIAs) (Castro & Timmis, 2002), Otimização por Colônia de Formigas (ACO, do inglês, *Ant Colony Optimization*) (Socha & Dorigo, 2008) e Algoritmos Genéticos (AGs) (Goldberg & Holland, 1988). O ajuste de parâmetros será realizado para as técnicas: Máquinas de Vetores de Suporte (SVMs, do inglês, *Support Vector Machines*) (Cristianini & Shawe-Taylor, 2000) e Redes Neurais Artificiais (RNs) (Haykin, 1999). Os classificadores são gerados para diferentes conjuntos de dados de expressão gênica. Além disso, conjuntos de dados do repositório da Universidade da Califórnia - Irvine (UCI, do inglês, University of California - Irvine) (Asuncion & Newman, 2007) foram utilizados em alguns experimentos. É importante utilizar mais de um conjunto de dados nas comparações para não estar sujeito à tendência de um conjunto em particular, como adverte Hoste & Daelemans (2005). O

desempenho obtido pelos classificadores gerados com o ajuste de parâmetros realizado pelos algoritmos bioinspirados são comparados empiricamente com outras abordagens.

Foram desenvolvidos três métodos experimentais para o ajuste de parâmetros. Os três métodos representam uma evolução nos métodos experimentais para se obter estimativas de desempenho de classificação confiáveis e evitar o uso indireto dos dados do conjunto de teste no processo de busca dos algoritmos bioinspirados. O primeiro método desenvolvido, denominado de Método Experimental A, foi aplicado apenas à SVMs e o segundo, denominado Método Experimental B, apenas à RNs. Os dois métodos possuíam viés no processo de ajuste, pois usavam indiretamente dados utilizados na estimativa do erro real do classificador. Portanto, um novo método foi proposto. Este, utiliza um procedimento de validação cruzada aninhado que permite realizar o ajuste de parâmetros e a estimação final do classificador simultaneamente, sem o viés existente nos outros dois métodos. Porém, há a desvantagem de ter um custo computacional mais elevado.

Nos experimentos com os três métodos desenvolvidos, foram realizadas comparações utilizando os algoritmos bioinspirados implementados e duas abordagens de referência. A abordagem Padrão, utilizada em todos os experimentos, usa valores estabelecidos pela implementação da técnica de AM para os parâmetros livres. A outra abordagem, utilizada no método experimental A, usa um algoritmo de ajuste da biblioteca que implementa as SVMs. Esse algoritmo, denominado *grid search*, realiza uma busca paralela em uma grade. Para os métodos experimentais B e C, além da Padrão, é usada uma abordagem denominada Aleatória, que gera um determinado número de valores aleatórios para os parâmetros, dentro de intervalos definidos. O número de valores gerados é aproximadamente igual ao número de avaliações que os algoritmos bioinspirados realizam da função de aptidão.

1.4 Principais Resultados Experimentais

No método experimental A, os algoritmos bioinspirados conseguiram as menores taxas de erro de validação (processo de estimação do erro de teste utilizado no ajuste de parâmetros) para três, dos quatro conjuntos de dados de expressão gênica utilizados. Porém, a melhoria obtida na taxa de erro de validação não foi refletida em melhoria para o erro de teste. Isso foi comprovado por meio do teste estatístico de Friedman (Demšar, 2006), que apontou que os algoritmos e as abordagens possuem desempenhos estatisticamente semelhantes.

Para os métodos experimentais B e C, os valores de parâmetros foram discretizados para um intervalo definido e os resultados foram armazenados em uma estrutura de dados apropriada. Dessa maneira, é possível extrair informações que auxiliam na análise dos resultados obtidos, como a correlação entre as taxas de erro de validação e de teste. Para o método experimental B, essas taxas são altamente correlacionadas para os quatro

conjuntos de dados utilizados, o que indica que a taxa de erro de validação é uma boa estimativa para os algoritmos bioinspirados guiarem suas buscas. O desempenho obtido pelos classificadores gerados com os parâmetros ajustados pelos algoritmos bioinspirados foi melhor do que as abordagens de referência. O algoritmo ACO conseguiu a menor taxa de erro de validação para três, dos quatro conjuntos de dados. Porém, essa melhoria foi pequena e não refletiu em melhorias para as taxas de erro de teste. O teste de Friedman encontrou diferenças significativas apenas para as taxas de erro de validação, em que o algoritmo ACO foi melhor do que a abordagem Padrão. Outras diferenças significativas para o conjunto de validação foram encontradas com o teste- t pareado corrigido.

O ajuste de parâmetros com o método experimental C foi realizado para RNs e SVMs. Para os experimentos com RNs, foram utilizados quatro conjuntos de dados de expressão gênica. O ACO conseguiu as menores taxas de erro de validação para os quatro conjuntos. Porém, isso não se refletiu para as taxas de erro de teste, em que diferentes algoritmos e abordagens conseguiram os menores valores. O teste de Friedman encontrou diferenças significativas apenas para as taxas de erro de validação, em que o ACO e a abordagem Aleatória foram melhores do que a abordagem Padrão. Para os experimentos com SVMs, foram utilizados seis conjuntos de dados de expressão gênica e quatro conjuntos do repositório UCI. Em geral, o algoritmo PSO e a abordagem Padrão conseguiram as menores taxas de erro de validação e de teste. A correlação para conjuntos de dados com muitos exemplos foi maior do que para pequenos conjuntos. Além disso, as correlações entre as taxas de erro de teste e de validação foram maiores para as SVMs do que para as RNs. O teste- t pareado corrigido encontrou diferenças significativas para algumas comparações e conjuntos de dados. O teste de Friedman apenas detectou diferenças significativas para as taxas de erro de validação. O algoritmo PSO e a abordagem Aleatória foram melhores que a abordagem Padrão e o algoritmo AG. Os algoritmos CSA e ACO também foram melhores que a abordagem Padrão.

Na comparação entre RNs e SVMs para classificação de dados de expressão gênica, as SVMs conseguiram os melhores resultados para os quatro conjuntos de dados. A menor diferença entre as taxas de erro foi de 0.7%, para o conjunto Glioma, e a maior de 7.2%, para o conjunto Leucemia. Devido ao viés de seleção de atributos nos experimentos realizados, não é possível obter conclusões com a comparação realizada. Porém, os resultados apontam para o melhor desempenho das SVMs.

1.5 Organização do Texto

No Capítulo 2 são introduzidos os principais conceitos de AM necessários para a compreensão deste trabalho. O desempenho das técnicas de AM será avaliado, principalmente, por meio da classificação de dados de expressão gênica. Assim, na primeira seção, é apresentado o problema de classificação e a avaliação de classificadores. Nas duas seções

seguintes são introduzidas as técnicas de classificação SVMs e RNs. Na próxima seção são apresentados o problema de seleção de modelo e de ajuste de parâmetros. Na última seção são feitas algumas considerações finais sobre o conteúdo abordado.

No Capítulo 3 a Computação Natural é introduzida; área à qual pertencem os algoritmos bioinspirados que serão utilizados no ajuste dos valores dos parâmetros livres das técnicas de AM apresentadas no Capítulo 2. Nas seções seguintes são apresentados os quatro algoritmos bioinspirados utilizados, ou seja, ACO, PSO, CSA e AGs. Na última seção são feitas algumas as considerações finais sobre esse capítulo.

No Capítulo 4 são descritos os experimentos para funções de *benchmark*. Antes de utilizar o algoritmos bioinspirados implementados para o ajuste de parâmetros, eles foram aplicados às funções de *benchmark* para verificar suas funcionalidades. Primeiramente são descritas as funções utilizadas. Em seguida, são apresentados os algoritmos implementados e seus parâmetros. Na próxima seção são apresentados os resultados obtidos. Na última seção, são feitas algumas considerações finais.

No Capítulo 5 são apresentados os experimentos realizados para ajuste de parâmetros de SVMs e RNs. Inicialmente, algumas características dos conjuntos de dados de expressão gênica e do repositório da UCI (Asuncion & Newman, 2007) são apresentados. Posteriormente, são apresentados os algoritmos bioinspirados utilizados nos experimentos realizados. Em seguida, são apresentados os pacotes (bibliotecas) que implementam as técnicas de AM investigadas. Na Seção seguinte, são descritos os métodos e as métricas empregadas para avaliar o desempenho dos classificadores. Nas próximas três seções são apresentados os diferentes métodos experimentais, que representam a evolução do procedimento experimental no sentido de buscar um modelo com menor viés, assim como os resultados obtidos.

No Capítulo 6 são apresentados e discutidos os principais resultados obtidos, as contribuições e as limitações deste trabalho e algumas direções para trabalhos futuros.

Capítulo 2

Aprendizado de Máquina

Aprendizado de Máquina é uma área baseada em IC e Estatística que estuda como desenvolver sistemas capazes de adquirir conhecimento de forma automática, ou seja, sistemas que melhoram automaticamente com a experiência. O modo de obtenção do conhecimento é o que leva pesquisadores dessa área a desenvolverem técnicas computacionais que tornem possível o aprendizado (Monard & Baranauskas, 2003; Mitchell, 1997).

Ainda está longe a possibilidade de tornar as máquinas tão hábeis de aprender quanto os humanos. Porém, algoritmos para tarefas específicas têm obtido grande sucesso, como no reconhecimento da fala, na mineração de dados, no reconhecimento de padrões e muitas outras (Mitchell, 1997).

Muitas técnicas de AM utilizam conhecimentos da Inteligência Computacional e da Estatística para construção de modelos capazes de inferir a partir de um conjunto de dados. Os exemplos desse conjunto são chamados de dados de treinamento e a partir desses faz-se a inferência indutiva, podendo gerar hipóteses verdadeiras ou não (Alpaydin, 2004; Monard & Baranauskas, 2003). Todo algoritmo de aprendizado, ou algoritmo de indução, ou simplesmente indutor, possui um viés (*bias*) indutivo, que é a preferência de uma hipótese sobre outra, que não são igualmente prováveis. Os exemplos de um conjunto de dados são formados por atributos e cada atributo especifica uma característica particular para esse conjunto. O aprendizado indutivo pode ser dividido em aprendizado supervisionado e não-supervisionado. Atualmente o aprendizado parcialmente supervisionado também tem atraído considerável atenção da comunidade de AM (Chapelle et al., 2006).

No aprendizado supervisionado, cada exemplo apresentado ao algoritmo de aprendizado possui um atributo especial que especifica o rótulo da classe real à qual este exemplo pertence. Se os rótulos das classes forem discretos, o problema é conhecido como classificação; se forem contínuos, como regressão, também conhecido como aproximação de funções.

No aprendizado não-supervisionado ou agrupamento, o algoritmo de aprendizado não tem conhecimento dos rótulos das classes reais. Dessa maneira, o algoritmo agrupa os exemplos por meio de extração de padrões de seus atributos.

No aprendizado parcialmente supervisionado, exemplos em que os rótulos das classes são conhecidos e também exemplos em que os rótulos das classes não são conhecidos são apresentados ao algoritmo de aprendizado. O algoritmo utiliza todos os exemplos rotulados e não rotulados durante o aprendizado.

Para avaliar o desempenho obtido com o ajuste de parâmetros as técnicas de AM serão aplicadas a problemas de classificação de dados de Bioinformática, mais especificamente de expressão gênica. Portanto na Seção 2.1 é descrito o problema de classificação, suas aplicações na área de Bioinformática e a avaliação de classificadores. Nas seções 2.2 e 2.3 serão apresentadas as técnicas de AM: SVMs e RNs, respectivamente, para as quais este trabalho investigou a utilização de técnicas para o ajuste de parâmetros. Em seguida, na Seção 2.4, o problema de seleção de modelo é investigado e alguns trabalhos publicados na área são apresentados. Finalizando, na Seção 2.5 são feitas algumas considerações finais.

2.1 Classificação

Em um problema de classificação os dados de entrada podem ser descritos pelo par ordenado (X, y) , em que X é um vetor que representa os atributos de um exemplo, $X = (x_1, x_2, x_3, \dots, x_n)$, e y é o rótulo da classe à qual este exemplo pertence. Na Tabela 2.1 tem-se um conjunto de dados para classificação do estado de saúde de um paciente. Nessa tabela cada linha representa um exemplo do conjunto de dados e cada coluna um atributo deste exemplo. O atributo *Diagnóstico* é especial, pois possui o rótulo da classe para cada exemplo, ou seja, doente ou saudável.

Tabela 2.1: Conjunto de dados para o diagnóstico da saúde de pacientes.

Exemplo	Febre	Enjôo	Manchas	Dor	Diagnóstico
T1	sim	sim	pequenas	sim	doente
T2	não	não	grandes	não	saudável
T3	sim	sim	pequenas	não	saudável
T4	sim	não	grandes	sim	doente
T5	sim	não	pequenas	sim	saudável
T6	não	não	grandes	sim	doente

Técnicas de AM podem ser utilizadas em problemas de classificação e cada uma utiliza um algoritmo de aprendizado para construir um modelo (classificador) que relaciona os atributos e os rótulos das classes. Um ponto importante dos algoritmos de aprendizado é construir modelos que possuam boa capacidade de generalização, ou seja, consigam prever, com alta taxa de acerto, rótulos das classes para exemplos que não foram apresentados anteriormente (Tan et al., 2005).

Outra característica dos classificadores a ser observada é em relação a interpretabilidade proporcionada ao ser humano. Os classificadores do tipo caixa-preta são aqueles

que possuem uma representação interna que, geralmente, não se consegue interpretar facilmente, ou seja, é difícil conhecer as características do problema que o levaram a uma determinada dedução. Um exemplo de classificador desse tipo são as RNs. Os classificadores orientados a conhecimento, como as Árvores de Decisão (ADs) e sistemas *fuzzy*, criam estruturas simbólicas que normalmente são mais compreensíveis do que os classificadores do tipo caixa-preta (Monard & Baranauskas, 2003).

Na Figura 2.1 está ilustrado um diagrama do processo de indução de um classificador e posteriormente a sua utilização. Primeiro o conjunto de treinamento, no qual os rótulos das classes dos exemplos são conhecidos, é utilizado por um algoritmo de aprendizado para construir um modelo. Após a construção, esse classificador pode ser aplicado para prever os rótulos das classes para exemplos do conjunto de teste.

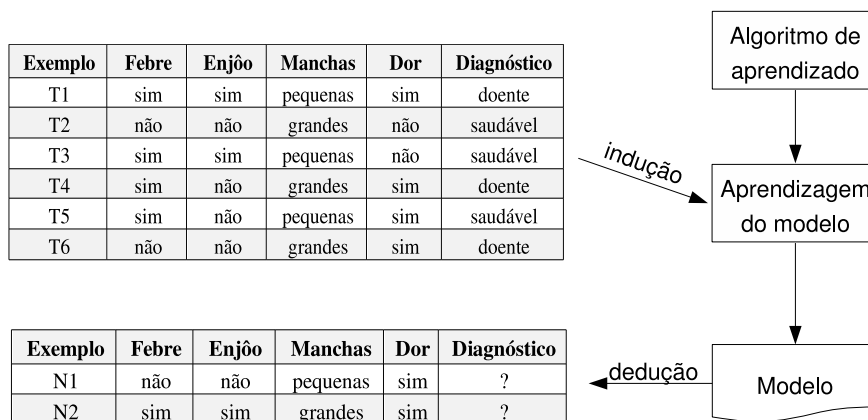


Figura 2.1: Diagrama do processo de indução de um classificador e sua utilização na dedução de novos exemplos.

2.1.1 Problemas de Classificação em Bioinformática

Entender as relações e interações entre os atributos de um conjunto de dados pode ser muito complicado. Isso pode acontecer, por exemplo, com dados biológicos que, geralmente, possuem muitos atributos. A classificação de dados da área de Bioinformática auxilia no entendimento das condições e das características de vários objetos biológicos. Um problema que tem sido muito tratado, por exemplo, é o de classificar células como normais e tumorais (Bajcsy et al., 2005).

Muitos bancos de dados de proteínas atualmente mantêm algum tipo de classificação destas seqüências, como o SCOP¹ (*do inglês, Structural Classification of Protein*), PIR² (*do inglês, Protein Information Resource*) e o PDB³ (*do inglês, Protein Data Bank*). Esses bancos podem possuir mais de um tipo de classificação para as proteínas armazenadas.

¹<http://scop.mrc-lmb.cam.ac.uk/scop/>

²<http://pir.georgetown.edu/pirwww/index.shtml>

³<http://www.pdb.org>

Devido aos projetos genoma, uma grande quantidade de dados biológicos estão sendo criados, sendo impraticável a classificação ou organização desses dados por especialistas. Diferentes classificadores têm sido usados para este problema, como SVMs, classificadores Bayesianos, k -vizinhos mais próximos, RNs e outros (Bajcsy et al., 2005). Uma descrição de diferentes técnicas computacionais para a classificação na área de Bioinformática pode ser vista em Larranaga et al. (2006).

2.1.2 Avaliação de Classificadores

Apesar do potencial de diferentes técnicas de AM, não há um algoritmo único que apresente o melhor desempenho para todos os problemas. Um ponto importante ao avaliar diferentes técnicas de AM é o método de amostragem dos dados que serão usados para a indução do classificador. O princípio dos métodos de amostragem é a formação de conjuntos disjuntos para treinamento e teste, ou seja, um exemplo deve pertencer ao conjunto de treinamento, utilizado para o aprendizado, ou ao conjunto de teste, utilizado para avaliar o desempenho do classificador, e nunca em ambos. O problema de se utilizar os mesmos dados para treinamento e teste é que o classificador produzido, em geral, não apresentará desempenho similar para dados não conhecidos. Assim, os métodos de amostragem auxiliam na obtenção de uma estimativa de erro do classificador mais fiel. A seguir são apresentados alguns métodos de amostragem (Monard & Baranauskas, 2003):

Holdout: Este estimador divide os exemplos de um conjunto de dados em um uma porcentagem fixa p para treinamento e $(1 - p)$ para teste. Normalmente $p > 1/2$, sendo $p = 2/3$ um valor muito usado;

Amostragem Aleatória: Na amostragem aleatória são gerados H conjuntos de treinamento e teste aleatoriamente e então, são induzidas H hipóteses a partir de cada conjunto de treinamento. O erro final é a média dos erros das H hipóteses para os conjuntos de testes independentes;

Validação Cruzada (CV, do inglês, *Cross-Validation*): Este método divide os exemplos em partições disjuntas, chamadas de *folds*. O número de *folds* utilizados pode variar conforme a quantidade de exemplos e a proporção de exemplos em cada classe. Para *k-fold cross-validation* o conjunto de dados é dividido em k *folds*, sendo que $(k - 1)$ são usados para treinamento e o *fold* restante para teste. Este processo se repete k vezes até que todos os *folds* tenham sido utilizados para teste. No caso da validação cruzada estratificada a proporção de exemplos em cada uma das classes é mantida para cada *fold* gerado.

Leave-One-Out: Este estimador é um caso específico do *k-fold cross-validation* em que $k =$ número de exemplos. Assim $(k - 1)$ exemplos são utilizados para treinar o

classificador e apenas um exemplo é utilizado para teste. Esse método tem alto custo computacional, já que esse processo se repete k vezes, para que todos os exemplos sejam usados uma vez para teste.

Bootstrap: Neste método repete-se o processo de classificação diversas vezes. Em cada experimento um novo conjunto de treinamento é obtido por amostragem com reposição a partir dos exemplos originais.

Uma vez escolhido um método de amostragem, deve-se definir o método para avaliar o desempenho dos classificadores. Uma maneira é basear-se no número correto e incorreto de predições realizadas pelo classificador. Isso é feito organizando esses números em uma tabela, chamada matriz de confusão. Na Tabela 2.2 é ilustrado um exemplo dessa matriz para um problema de classificação binária, ou seja, os exemplos são classificados como pertencentes à classe positiva ou negativa. Nessa tabela as linhas indicam a classe verdadeira e as colunas indicam a classe predita.

Tabela 2.2: Matriz de confusão.

	Classe Predita	
Classe Verdadeira	Positiva	Negativa
Positiva	VP	FN
Negativa	FP	VN

As siglas utilizadas na Tabela 2.2 (VP, VN, FP e FN) são descritas a seguir:

- Verdadeiro Positivo (VP): total de exemplos preditos como pertencentes à classe positiva e que realmente pertencem à essa classe;
- Verdadeiro Negativo (VN): total de exemplos preditos como pertencentes à classe negativa e que realmente pertencem à essa classe;
- Falso Positivo (FP): total de exemplos preditos como pertencentes à classe positiva mas que na verdade pertencem à classe negativa;
- Falso Negativo (FN): total de exemplos preditos como pertencentes à classe negativa mas que na verdade pertencem à classe positiva.

A soma dos valores que aparecem na diagonal principal da Tabela 2.2 representa o número total de exemplos classificados corretamente. A soma dos valores que não se encontram na diagonal principal é igual ao número de exemplos classificados incorretamente. Utilizando a matriz de confusão é possível calcular a taxa de acerto e a taxa de erro, conforme as equações 2.1 e 2.2, respectivamente. A taxa de acerto indica a taxa de exemplos classificados corretamente e a taxa de erro, a taxa de exemplos classificados incorretamente.

$$\text{Taxa de acerto} = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.1)$$

$$\text{Taxa de erro} = 1 - \text{Taxa de acerto} \quad (2.2)$$

A Taxa de Verdadeiros Positivos (TVP), também conhecida como sensibilidade ou *recall*, estima a probabilidade de um exemplo pertencente à classe positiva ser predito como positivo. A especificidade estima a probabilidade de um exemplo pertencente à classe negativa ser predito corretamente como negativo. A Taxa de Falsos Positivos (TFP) estima a probabilidade de um exemplo pertencente à classe negativa ser predito como positivo. Essas três medidas são estimadas pelas equações 2.3, 2.4 e 2.5, respectivamente.

$$TVP = \frac{VP}{VP + FN} \quad (2.3)$$

$$\text{especificidade} = \frac{VN}{VN + FP} \quad (2.4)$$

$$TFP = \frac{FP}{FP + VN} \quad (2.5)$$

A medida de precisão, dada pela equação 2.6, estima a probabilidade da predição positiva estar correta. Uma outra medida, a Medida-F (*F-measure*), combina de maneira balanceada as medidas de precisão e *recall*. A medida-F é dada pela equação 2.7.

$$\text{precisão} = \frac{VP}{(VP + FP)} \quad (2.6)$$

$$\text{medida-F} = \frac{2}{1/\text{precisão} + 1/\text{recall}} \quad (2.7)$$

Atualmente, curvas ROC (do inglês, *Receiver Operating Characteristics*) têm sido muito utilizadas em AM para avaliar classificadores. Curvas ROC são gráficos bidimensionais em que a taxa de VP é plotada no eixo Y e a taxa de FP é plotada no eixo X. Um gráfico ROC apresenta um balanço entre custos (falsos positivos) e benefícios (verdadeiros positivos) (Fawcett, 2006). Uma curva ROC pode ser utilizada quando classificadores binários a serem comparados produzem como saída valores de probabilidade ou *scores*, denominados de classificadores probabilísticos (Fawcett, 2006). A curva ROC permite visualizar, organizar e selecionar classificadores binários baseados em seu desempenho. Uma única medida para descrever essa curva é a área abaixo da curva (AUC, do inglês, *Area Under Curve*). Essa medida pode variar de zero a um. Quanto maior a área, melhor é o classificador. A Figura 2.2 mostra um exemplo de uma curva ROC cuja AUC é 0.883.

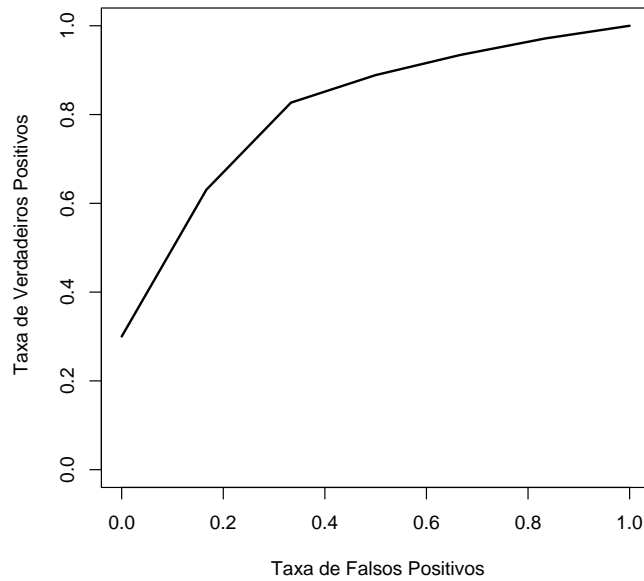


Figura 2.2: Exemplo de curva ROC.

Testes estatísticos

A taxa de acerto ou a taxa de erro pode ser utilizada para comparar dois algoritmos de aprendizado A e B utilizando um teste de hipóteses pareado (Mitchell, 1997). Com esse teste, é possível avaliar se houve diferença estatística significativa entre o desempenho dos algoritmos A e B para um determinado conjunto de dados. Seja um conjunto de dados particionado segundo o método de amostragem *k-fold cross-validation*. Considere agora $e_i(f_A)$ a taxa de erro (ou de acerto) obtido por um classificador f_A induzido pelo algoritmo de aprendizado A e $e_i(f_B)$ a taxa de erro (ou de acerto) obtido por um classificador f_B induzido pelo algoritmo de aprendizado B . Uma alternativa para aplicar um teste estatístico seria, primeiro, determinar as diferenças entre os erros obtidos por cada classificador, para cada partição de teste i , segundo a Equação 2.8.

$$dif_i = e_i(f_A) - e_i(f_B) \quad (2.8)$$

Com as diferenças entre cada partição de teste, calcular a média e o desvio padrão segundo as equações 2.9 e 2.10, respectivamente.

$$\bar{m}_d = \frac{1}{k} \sum_{i=1}^k dif_i \quad (2.9)$$

$$s_d = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (dif_i - \bar{m}_d)^2} \quad (2.10)$$

A hipótese nula é de que não há diferença estatística entre os resultados obtidos pelos algoritmos A e B . Assim, calcula-se o valor de teste t obtido conforme a equação 2.11. Esse teste é chamado de teste- t pareado corrigido e foi proposto por Nadeau & Bengio (2003).

$$t_{obt} = \frac{\bar{m}_d}{\sqrt{\left(\frac{1}{k} + \frac{1}{k-1}\right)s_d^2}} \quad (2.11)$$

A hipótese nula é rejeitada se t_{obt} for maior que $t_{k-1, 1-\frac{\alpha}{2}}$, em que t_{k-1} é a distribuição t de *Student* com $k-1$ graus de liberdade e α corresponde ao nível de significância do teste. Assim pode-se afirmar se os resultados obtidos pelos algoritmos A e B são estatisticamente diferentes com um nível de significância α .

O teste- t pareado corrigido não pode ser aplicado para múltiplos algoritmos e múltiplos conjuntos de dados. Para esses tipos de comparação, o teste não-paramétrico de Friedman (Demšar, 2006) pode ser usado. Os testes não-paramétricos não exigem que as distribuições da variável na população sejam conhecidas. Com esse teste, é possível verificar se há ou não diferenças significativas entre os diferentes classificadores para todos os conjuntos de dados. Porém, o teste de Friedman não determina quais são os algoritmos significativamente diferentes. Portanto, caso o teste de Friedman rejeite a hipótese nula, ou seja, rejeite a hipótese de que não há diferença significativa entre os classificadores, o teste *à posteriori* Nemenyi (Demšar, 2006) pode ser aplicado para verificar onde ocorrem as diferenças. O desempenho de dois classificadores é significativamente diferente se o *ranking*-médio de Friedman diferir por pelo menos uma diferença crítica. Essa diferença é calculada com base no número de classificadores, número de algoritmos e na distribuição de *Student* (Demšar, 2006).

2.2 Máquinas de Vetores de Suporte

As SVMs são baseadas na Teoria do Aprendizado Estatístico (Vapnik & Chervonenkis, 1971), mais especificamente, na Teoria de Minimização do Risco Estrutural (Vapnik, 1995). Os vetores de suporte utilizados pelas SVMs são exemplos que estão próximos da superfície de decisão e portanto, são os mais difíceis de serem classificados (Haykin, 1999). São esses exemplos que influenciam diretamente na localização da superfície de decisão.

As SVMs vêm sendo utilizadas com sucesso em diversos problemas de Bioinformática, como a classificação de dados de expressão gênica e a detecção de homologia entre proteínas (Cristianini & Shawe-Taylor, 2000). Entre as características que a tornaram popular na Bioinformática estão a boa capacidade de generalização e a robustez em lidar com dados de grande dimensão (Souto et al., 2003).

Para o problema de classificação, o princípio das SVMs está em encontrar um hiperplano ótimo que separa satisfatoriamente os dados de entrada. O hiperplano ótimo

é definido como aquele para o qual a margem de separação entre as classes é maximizada (Haykin, 1999). Utilizando o hiperplano as SVMs são capazes de classificar apenas conjuntos de entrada linearmente separáveis. Um conjunto de dados é linearmente separável quando é possível separar as diferentes classes com um hiperplano (Mitchell, 1997). Um hiperplano ótimo e os vetores de suporte para exemplos linearmente separáveis estão ilustrados na Figura 2.3.

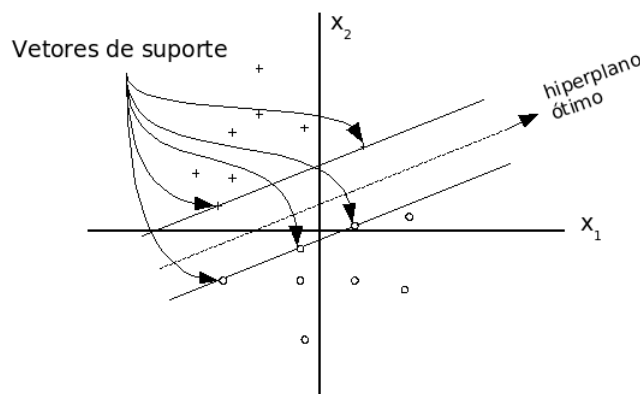


Figura 2.3: A idéia do hiperplano ótimo para exemplos linearmente separáveis.

Quando essas classes não podem ser separadas satisfatoriamente por um hiperplano, as SVMs devem ser generalizadas. Isso pode ser feito com o uso de funções de mapeamento Φ_1, \dots, Φ_M no domínio do espaço de entrada. Por meio destas funções, cada exemplo do conjunto de treinamento é mapeado para um novo espaço, o espaço de características. Neste espaço os exemplos podem se tornar linearmente separáveis e então um hiperplano ótimo pode ser encontrado (Haykin, 1999).

A escolha da função de mapeamento apropriada varia com o espaço de características estudado. A única informação necessária sobre o mapeamento é a definição de como o produto interno $\Phi(x_i) \cdot \Phi(x_j)$ pode ser realizado para cada par de exemplos x_i e x_j (Souto et al., 2003). Essa informação é obtida com os *kernels*: funções que têm como entrada dois pontos x_i e x_j do espaço de entrada e calculam o produto escalar $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ no espaço de características (Haykin, 1999).

Dentre os *kernels* mais utilizados estão o linear, o polinomial, o de funções de base radial (RBF, do inglês, *Radial Basis Function*) ou gaussiano e o sigmoideal. Cada *kernel* possui um conjunto de parâmetros livres, ou seja, parâmetros cujos valores devem ser especificados pelo usuário. Na Tabela 2.3 são apresentados esses *kernels* e alguns de seus parâmetros. Além disso um parâmetro das SVMs, denominado custo, precisa ser especificado. Esse é um parâmetro regulador que controla o equilíbrio entre a complexidade do modelo e o número de exemplos não-separáveis (Haykin, 1999).

Tabela 2.3: Principais *kernels* utilizados nas SVMs (Haykin, 1999).

<i>Kernels</i>	Função $K(x_i, x_j)$ correspondente	Parâmetros
polinomial	$(x_i^T x_j + 1)^p$	A potência p deve ser especificada pelo usuário
RBF	$\exp(-\frac{1}{2\sigma^2} \ x_i - x_j\ ^2)$	A amplitude σ^2 é especificada pelo usuário
sigmoidal	$\tanh(\beta_0 x_i x_j + \beta_1)$	Satisfeito apenas para alguns valores β_0 e β_1

2.3 Redes Neurais Artificiais

As RNs são inspiradas no cérebro e no sistema nervoso. Ambos são sistemas complexos, paralelamente distribuídos e não-lineares. As RNs são compostas por unidades de processamento simples, denominados neurônios ou nós, dispostos em uma ou mais camadas e interligados por meio de conexões.

As RNs têm sido aplicadas em uma grande variedade de domínios, entre eles, a Bioinformática (Narayanan et al., 2002). Segundo Seiffert et al. (2006), o comportamento robusto das RNs frente a dados com ruídos e a alta adaptabilidade fornecem muitos dos requisitos necessários dessa área. Exemplos de aplicações de RNs em problemas de Bioinformática são os trabalhos de Wang et al. (2000), que utiliza RNs para classificação de seqüências de proteínas, e Draghici & Potter (2003), que usou RNs para prever a resistência do vírus HIV a drogas.

O primeiro modelo formal para um neurônio artificial foi desenvolvido por McCulloch & Pitts (1943), que mostraram que seu modelo era capaz de realizar, a princípio, a computação de qualquer função computável utilizando um número suficiente de neurônios artificiais e pesos sinápticos ajustáveis (Haykin, 1999). Os neurônios artificiais são unidades de processamento de informação que realizam um trabalho relativamente simples: recebem entradas de outros neurônios ou do meio externo e usam essas entradas para computar um sinal de saída que é propagado para outras unidades ou para o meio externo.

Cada unidade é composta basicamente por três elementos (Haykin, 1999):

- **Sinapse** - peso acoplado a cada entrada do neurônio que determina se a entrada terá efeito inibitório ou excitatório, resultando na ativação ou não do neurônio;
- **Somador** ou *net input* - realiza a soma ponderada entre as entradas e seus respectivos pesos;
- **Função de Ativação** - determina o nível de ativação baseado no resultado produzido pelo somador, limitando o valor de saída de cada neurônio.

Na Figura 2.4 é apresentado o esquema de um neurônio artificial e de uma rede neural do tipo *feed-forward*, ou seja o fluxo de processamento é estritamente da entrada para a saída.

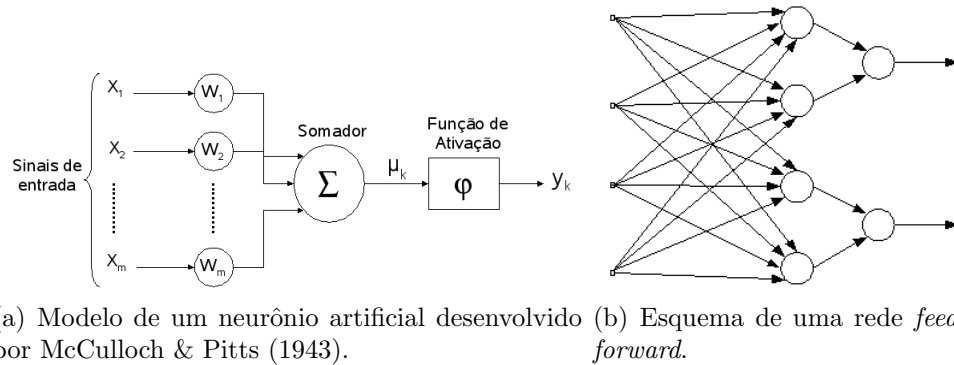


Figura 2.4: Neurônio artificial e o esquema de uma rede *feed-forward*.

Diversas funções podem ser utilizadas como funções de ativação e devem ser especificadas pelo usuário. A escolha depende da especificação do problema que o neurônio se propõe a resolver (Hagan et al., 1996). As funções logística, de limiar e tangente hiperbólica são três exemplos de funções de ativação muito utilizadas. Segundo Basheer & Hajmeer (2000) uma escolha ótima de outros parâmetros, como o tamanho da rede e a taxa de aprendizagem, afetam o desempenho final da rede.

Os pesos associados a cada conexão entre os neurônios (sinapse) são responsáveis pelo armazenamento do conhecimento. Um algoritmo de aprendizado de RNs deve ser capaz de atribuir pesos a essas conexões durante o processo de treinamento, de maneira que a rede neural seja capaz de classificar corretamente o maior número de exemplos possíveis do conjunto de treinamento e seja capaz de generalizar para novos exemplos.

2.4 Seleção de Modelo e Ajuste de Parâmetros

Diversos modelos podem apresentar bom desempenho preditivo quando utilizados para resolver um determinado problema de classificação. O problema de escolher o melhor entre os diversos modelos disponíveis é chamado de seleção de modelo. Para todos os casos, o objetivo é minimizar o erro de generalização, pois minimizar o erro de treinamento pode causar *overfitting*, ou seja, o modelo gerado pode memorizar os exemplos do conjunto de treinamento ao invés de extrair suas características gerais, o que, por exemplo, prejudica o desempenho para novos exemplos (Kearns et al., 1997). Na prática, o problema de seleção de modelo pode ser reduzido ao ajuste dos valores de parâmetros quando se tem algum conhecimento prévio do problema. Na Seção 2.4, o problema de ajuste de parâmetros será discutido com maiores detalhes e alguns trabalhos serão apresentados.

Uma técnica para seleção de modelo de SVMs envolve a minimização do erro de gene-

realização usando informações do gradiente sobre o conjunto de parâmetros. Porém, essa abordagem apresenta problemas, como a necessidade de funções diferenciáveis e o alto risco de atingir um mínimo local (Chapelle et al., 2002). Os AGs foram usados por Xu & Chan (2003) para a seleção de modelo para SVMs multi-classes. Os resultados para os problemas investigados mostraram que o método melhora o desempenho de classificação com complexidade aceitável.

Em Anders & Korn (1999), os autores utilizam diferentes estratégias baseadas em conceitos estatísticos para a seleção de modelo para RNs, como teste de hipóteses, critérios de informação e CV. Com os resultados obtidos nesse trabalho, verificou-se que o teste de hipótese de Teräsvirta et al. (1993) foi a estratégia mais estável.

Para a seleção de modelos em ADs, Yildiz & Alpaydin (2005) utilizaram diferentes abordagens, como critérios de informação Akaike (AIC, do inglês, *Akaike Information Criterion* (Akaike, 1974) e Bayesianos (Schwarz, 1978), além da CV. Este último produziu árvores mais simples que os outros dois métodos, com a desvantagem de um maior tempo de treinamento.

Estudos de diversos métodos de seleção de modelo podem ser encontrados em Kearns et al. (1997) e Bartlett et al. (2002).

Ajuste de Parâmetros

Grande parte dos algoritmos de AM possuem parâmetros cujos valores devem ser especificados pelo usuário. Esses valores para os parâmetros livres, em geral, influenciam diretamente no desempenho de modelos induzidos (Kohavi & John, 1995), o que pode ser entendido como uma deficiência das técnicas de AM. O objetivo do processo de ajuste desses parâmetros pode ser visto como encontrar os melhores valores dos parâmetros livres para um determinado conjunto de dados (Hutter & Hamadi, 2005). O ajuste de parâmetros com o intuito de obter melhor desempenho e maior robustez dessas técnicas é uma tarefa subjetiva, que pode consumir muito tempo.

Para definir um conjunto de valores, as atuais técnicas de ajuste normalmente consideram a interação entre o viés (*bias*) do algoritmo de indução (Mitchell, 1982) e o conjunto de treinamento disponível. Técnicas comuns, como a busca exaustiva, são intratáveis quando há mais do que dois parâmetros para serem ajustados (Chapelle et al., 2002). Heurísticas podem ser aplicadas com relativo sucesso para uma variedade de conjuntos de dados. Porém, a melhor estratégia é obter valores dos parâmetros que funcionem bem para o conjunto particular de dados que está sendo analisado (Kohavi & John, 1995). A abordagem mais comum para definir os valores dos parâmetros livres é por tentativa e erro, a qual é altamente subjetiva. Além disso, a busca pelos melhores valores para os parâmetros, geralmente, envolve a otimização por um grande espaço de busca, o que torna esse problema muito custoso computacionalmente.

Por essa razão, técnicas alternativas de otimização têm sido aplicadas para ajustar de forma automática os parâmetros livres de algoritmos de AM. Algumas dessas técnicas têm obtido resultados promissores, como AGs, *Simulated Annealing* (Kirkpatrick et al., 1983), *Tabu Search* (Glover & Laguna, 1997) e PSO.

Ajuste de Parâmetros de SVMs

O desempenho das SVMs é diretamente influenciado pela escolha da função de *kernel* e os valores de seus parâmetros (Chapelle et al., 2002). Como dito anteriormente, essa dependência dos valores dos parâmetros não é uma exclusividade das SVMs. Muitos algoritmos de aprendizado dependem de uma escolha adequada para encontrarem um bom modelo. O conhecimento sobre o domínio pode auxiliar na escolha da função de *kernel* apropriada, reduzindo o problema de seleção de modelo para o ajuste de parâmetros (Cristianini & Shawe-Taylor, 2000).

Para o ajuste de parâmetros de SVMs, métodos de otimização baseados no gradiente são muito utilizados. Porém, segundo Imbault & Lebart (2004), esses métodos não resolvem totalmente o problema, a menos que um ponto inicial seja conhecido. Os autores mostraram que o problema de ajuste de parâmetros apresenta mínimo local e compararam métodos clássicos de ajuste que utilizam busca local com AGs e *Simulated Annealing*, que são métodos de minimização global, sendo o primeiro, uma técnica bioinspirada. Um fator importante dos métodos de minimização global é que esses são desenvolvidos para evitar mínimos locais. Os resultados mostraram que os dois métodos citados obtiveram soluções próximas da ótima de forma mais robusta e eficiente.

Algoritmos Genéticos também foram utilizados por Lorena & Carvalho (2006) para ajustar os parâmetros de SVMs multiclases com *kernel* gaussiano. Os autores obtiveram maiores taxas de acerto utilizando AGs se comparado aos resultados obtidos utilizando-se valores fixos para os parâmetros. Esse resultado ocorreu para as quatro bases de dados testadas. Huang & Wang (2006) e Souza & Carvalho (2005) utilizaram AGs para selecionar características (atributos) de conjuntos de dados e ajustar parâmetros de SVMs simultaneamente. No primeiro caso, foram realizados testes utilizando vários conjuntos de dados e a abordagem baseada em AGs teve boa taxa de acerto se comparada com a técnica *grid search*. No segundo caso, foi utilizado um conjunto de dados de expressão gênica e os resultados obtidos foram equivalentes a outros encontrados na literatura.

Em Souza et al. (2006), a técnica PSO foi utilizada para ajustar parâmetros de SVMs multiclases com *kernel* gaussiano. Foram testadas quatro bases de dados e os resultados obtidos foram comparados com os obtidos por outras técnicas de ajuste. Essas técnicas foram denominadas *grid search*, que usa a busca exaustiva *Naive*, que usa os mesmos valores padrões da biblioteca LIBSVM (Chang & Lin, 2001) para todas as SVMs binárias, e *Global*, que usa um conjunto de validação para estimar o erro de generalização. As técnicas PSO e Global conseguiram os menores erros de classificação para uma base, enquanto

que a *grid search* obteve melhores resultados para as outras duas bases. Apesar disso, os resultados obtidos pela PSO ficaram próximos aos melhores alcançados e, segundo os autores, não foi possível determinar o melhor método para todas as bases de dados testadas.

Ajuste de Parâmetros de RNs

Algoritmos evolutivos são empregados para ajuste de parâmetros e treinamento de RNs desde o início da década de 90, como pode ser visto em Miller et al. (1989); Dodd (1990); Hintz & Spofford (1990); Braun & Weisbrod (1993), e ainda são utilizados nos dias atuais. Os trabalhos de Leung et al. (2003) e Tsai et al. (2006) propõem modificações nos AGs para a otimização automática e simultânea da topologia (estrutura) das RNs e dos parâmetros do algoritmo de treinamento. Em Leung et al. (2003), os autores realizaram experimentos com funções *benchmark*. Os resultados obtidos pelo AG proposto superaram os obtidos pelo AG padrão, que utilizou *crossover* aritmético e mutação não-uniforme. Duas aplicações foram testadas com as RNs para ilustrar o desempenho dos algoritmos propostos. Em ambas, o AG proposto superou o AG padrão no ajuste de parâmetros, e o algoritmo de aprendizado *backpropagation*, para o treinamento da rede. Tsai et al. (2006) aplicaram um algoritmo híbrido, que combina AG com o método *Taguchi* (Phadke, 1995), o qual chamaram de HTGA (do inglês, *hybrid Taguchi-Genetic Algorithm*), para ajustar a estrutura e os parâmetros de RNs. Os autores utilizaram as mesmas aplicações testadas em Leung et al. (2003) e compararam os resultados obtidos. O HTGA foi superior em relação ao valor de aptidão encontrado, enquanto que a complexidade da rede variou conforme a aplicação.

Um estudo comparativo entre sistemas evolutivos híbridos para geração e otimização da estrutura de uma RN multi-camadas foi realizado por Castillo et al. (2007). Esse estudo usou um método chamado *G-Prop* para otimizar o número de neurônios ocultos e os pesos iniciais das RNs e outro método chamado *ev-QP* para buscar pelos parâmetros de um algoritmo de treinamento. Utilizou-se também um algoritmo co-evolutivo, chamado *co-evolutive*, para tratar dos parâmetros referente à arquitetura, aos pesos iniciais e aos parâmetros do algoritmo de treinamento. Os resultados mostraram que os algoritmos evolutivos apresentaram melhor habilidade de classificação do que o QuickProp (Fahlman, 1988). O primeiro método (GProp) obteve pequenas taxas de erro de classificação. Porém, o tempo de processamento foi um pouco superior se comparado a outros métodos. O segundo método (ev-QP) teve o menor tempo de processamento, mas produziu as maiores taxas de erro. O método co-evolutivo obteve os menores tempo de processamento e, ao mesmo tempo, melhorou a habilidade de classificação.

Em Gao et al. (2006), a técnica PSO foi aplicada para ajustar simultaneamente a estrutura e os pesos das conexões de RNs. Os autores propuseram uma modificação no PSO, chamado de *SPSO*, para solucionar esse problema. Os resultados obtidos foram compara-

dos com o algoritmo *backpropagation* e com um AG desenvolvido para o mesmo propósito. As redes foram aplicadas para o problema de pontuação de crédito. O SPSO convergiu mais rapidamente e conseguiu maior acurácia do que o algoritmo *backpropagation*. O AG foi o que obteve as menores taxas de acurácia.

Um algoritmo baseado no sistema imunológico, denominado *Immune Algorithm* (IA), foi empregado por Kim (2002) para ajustar parâmetros de RNs *fuzzy*. O algoritmo foi utilizado para gerar regras *fuzzy* para estruturas de rede neural-*fuzzy*. Os autores relatam que por meio das simulações realizadas foi possível constatar a capacidade do IA em otimizar regras fuzzy.

2.5 Considerações Finais

Neste capítulo, foram abordados os principais conceitos de AM e as duas técnicas de AM que foram utilizadas durante este estudo: SVMs e RNs. Como essas técnicas foram aplicadas em tarefas de classificação, foram descritas brevemente os principais conceitos relacionados a esse tema e sua utilização em dados da área de Bioinformática.

Em seguida, foi abordado o problema da seleção de modelo e de ajuste de parâmetros para técnicas de AM. Alguns trabalhos publicados que utilizam técnicas bioinspiradas para o ajuste de parâmetros foram citados.

No próximo capítulo são apresentados os conceitos de Computação Natural e as técnicas bioinspiradas utilizadas neste estudo para o ajuste de parâmetros das técnicas de AM.

Capítulo 3

Computação Natural

Ao contrário da divisão das ciências atuais, que está cada vez mais sendo especializada, a Computação Natural utiliza o conhecimento de diversas áreas para melhorar a compreensão da vida e desenvolver novos paradigmas computacionais. A Computação Natural usa elementos da Computação, Biologia, Física, Engenharia e Química.

A Computação Natural ou Bioinspirada é formada por três frentes de estudo:

A natureza como fonte de inspiração para a computação: estuda como conceitos, princípios e mecanismos da natureza podem ser utilizados para a criação de novas técnicas computacionais. É possível observar na natureza soluções eficientes e elegantes para resolver uma grande variedade de problemas. Muitos desses problemas não podem ser solucionados por técnicas computacionais tradicionais, porém a natureza é capaz de resolvê-los facilmente. Como exemplo, podem ser citados problemas de otimização e de reconhecimento de padrões (Carvalho et al., 2004).

Simulação e emulação da natureza por meio da computação: é basicamente um processo de síntese com o objetivo de criar formas, padrões e comportamentos que reproduzam a vida como nós a conhecemos. Os resultados dos estudos dessa área podem ser utilizados para simular vários fenômenos naturais, possibilitando um maior entendimento da natureza (Castro, 2007).

Utilização de materiais naturais para executar computação: estuda o uso de materiais naturais, como moléculas de DNA, para realizar computação. Esses novos paradigmas computacionais podem substituir ou serem acrescentados aos materiais de silício usados atualmente para computar (Castro, 2007).

Este estudo envolve a primeira frente de estudo da Computação Natural, pois fez uso de técnicas bioinspiradas. As técnicas bioinspiradas têm sido formalizadas como metaheurísticas para problemas de otimização. Segundo Dorigo et al. (2006), metaheurística é um conjunto de conceitos algorítmicos que pode ser utilizado para definir métodos heurísticos aplicáveis a um amplo conjunto de problemas. Exemplos de metaheurísticas incluem

a busca tabu (Glover & Laguna, 1997) a busca local iterativa (Lourenço et al., 2002) e a computação evolutiva (Holland, 1975). Os algoritmos definidos como metaheurísticas, conforme Maniezzo et al. (2004), possuem algumas heurísticas básicas para escapar do mínimo local: iniciam a partir de uma solução nula e adicionam elementos até obter uma boa solução ou iniciam com uma solução completa e iterativamente modificam alguns elementos dessa até um critério de parada.

O ajuste dos valores dos parâmetros de técnicas de AM foram investigadas neste estudo com uso de técnicas bioinspiradas. Assim, as seções seguintes apresentam os principais conceitos relacionados às técnicas bioinspiradas. Na Seção 3.1 é apresentada uma introdução à inteligência de enxames e às técnicas ACO e PSO, pertencentes à esse paradigma. Na Seção 3.2 os SIAs são apresentados, com ênfase para o princípio da seleção clonal. Em seguida, na Seção 3.3 os conceitos referentes aos AGs são abordados. Por último, são feitas algumas considerações finais na Seção 3.4.

3.1 Inteligência de Enxames

A inteligência de enxames, ou inteligência coletiva, se refere a agentes (indivíduos) que apresentam um nível superior de inteligência dentro do comportamento social. Os indivíduos devem ser capazes de interagir entre si e com o ambiente. A vida social é interessante sob vários aspectos para algumas espécies. As vantagens podem ser uma maior facilidade de encontrar comida, divisão do trabalho, melhor aproveitamento de suas capacidades e até mesmo evitar predadores e facilitar a caça (Castro, 2006).

Os algoritmos de inteligência de enxames são compostos por indivíduos simples que atuam de forma auto-organizada, isto é, sem qualquer forma de controle central sobre os membros do enxame. Segundo Millonas (1994), os sistemas baseados em inteligência coletiva possuem cinco princípios:

proximidade: indivíduos de uma população devem interagir entre si;

qualidade: indivíduos devem ser capazes de avaliar a interação entre eles e com o ambiente;

diversidade: a capacidade de um sistema reagir contra ações inesperadas;

estabilidade: os indivíduos não podem modificar seu comportamento em resposta a todas modificações do ambiente;

adaptabilidade: os indivíduos devem ser capazes de se adaptar às mudanças do ambiente e da população.

As técnicas ACO e PSO são baseadas em sistemas de enxames. A primeira é baseada no comportamento de formigas em busca por alimento e a segunda na organização existente entre bandos de pássaros, peixes e no comportamento social humano.

3.1.1 Otimização por Colônia de Formigas

As formigas, de forma especial, têm servido de inspiração para um grande número de técnicas baseadas em inteligência de enxames. Dentre essas, a mais estudada e de maior sucesso é a técnica de otimização de propósito geral, conhecida como ACO (Dorigo et al., 2006). A ACO é inspirada no comportamento das formigas na busca por alimento. O principal aspecto desse comportamento é a comunicação que ocorre entre os indivíduos da colônia por meio do depósito de feromônio nas trilhas percorridas.

Inicialmente, as formigas exploram de maneira aleatória uma área em busca de alimento. Durante a movimentação, essas formigas depositam feromônio pelo solo ao longo do caminho percorrido (Blum, 2005). O feromônio é uma substância química cujo odor é sentido pelas formigas. Ao escolher um caminho dentre mais de uma opção, é provável que as formigas escolham o caminho marcado com uma maior concentração de feromônio. Como a probabilidade das formigas que alcançaram a comida pelo menor caminho retornarem antes das que escolheram o caminho mais longo é maior, o menor caminho ficará com uma maior concentração de feromônio e provavelmente será o caminho seguido pelas outras formigas. Na Figura 3.1 está ilustrada essa idéia, mostrando o aumento de formigas que utilizam o caminho mais curto entre o ninho e a fonte de alimento ao longo do tempo.

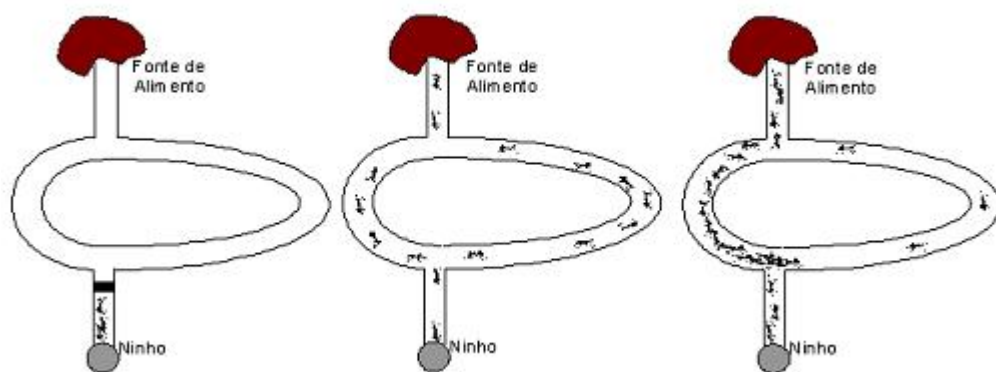


Figura 3.1: Formigas em busca de alimento. Ao passar do tempo, o menor caminho possuirá a maior quantidade de feromônio depositado. (Fonte: <http://1sin.unisantos.br/1vcon/1vcon>).

A ACO explora um mecanismo similar para resolver problemas de otimização e foi formalizado como metaheurística para problemas de otimização combinatorial por Dorigo & Di-Caro (1999).

O funcionamento da ACO pode ser resumido como um conjunto de agentes computacionais concorrentes e assíncronos (como uma colônia de formigas) que se movimentam através de estados do problema, que correspondem às soluções parciais. A movimentação dos agentes é baseada em dois parâmetros: trilha e atratividade. Durante seu movimento, cada formiga constrói incrementalmente uma solução para o problema. Durante a fase de construção ou finalização de uma solução, a formiga avalia a solução encontrada e modifica o valor de feromônio associado com as trilhas usadas. O total de feromônio nas trilhas direcionará a busca para as demais formigas (Maniezzo et al., 2004).

Um algoritmo ACO inclui dois mecanismos adicionais: evaporação da trilha e, opcionalmente, ações *daemon*. A evaporação da trilha decrementa a quantidade de feromônio em todas as trilhas com o passar do tempo com o intuito de evitar o acúmulo ilimitado de feromônio. Ações *daemons* podem ser utilizadas para permitir ações centralizadas, o que não acontece com as colônias naturais de formigas (Maniezzo et al., 2004).

Em ACO para problemas combinatoriais, formigas constroem soluções incrementalmente. Inicialmente cada formiga começa com uma solução vazia s^0 e a cada passo de construção i uma componente da solução é adicionada. A definição de componente da solução é dependente da aplicação. Para o problema do caixeiro viajante, por exemplo, uma componente da solução é uma cidade que é adicionada ao percurso. Para escolher qual componente da solução C^i deve ser adicionada a s^i , uma escolha probabilística é feita, considerando-se normalmente o total de feromônio τ associado com uma determinada componente e uma possível informação heurística sobre o problema (Socha, 2004).

A ACO foi desenvolvida inicialmente para resolver problemas de otimização combinatorial, logo a versão original não se aplica à problemas de otimização de domínio contínuo. Uma extensão para a ACO foi desenvolvida por Socha (2004), a qual permite a otimização de problemas com variáveis contínuas e mistas (discretas e contínuas). Posteriormente, Socha & Dorigo (2008) desenvolveram o $ACO_{\mathbb{R}}$ para problemas de domínios contínuos. A idéia principal do $ACO_{\mathbb{R}}$ é substituir a distribuição de probabilidade discreta utilizada no ACO convencional por uma contínua, isto é, por uma Função de Densidade de Probabilidade (FDP).

No Algoritmo 1 são descritos os passos do ACO para domínios contínuos. As soluções encontradas por uma formiga l é dada por s_l . Ao invés de escolher uma componente da solução como em problemas combinatoriais, uma formiga gera um número aleatório de acordo com uma FDP $P^i(x^i)$ no passo i .

3.1.2 Otimização por Enxame de Partículas

A PSO é uma técnica de otimização global desenvolvida por Kennedy & Eberhart (1995) e foi introduzida inicialmente para otimização de funções contínuas não lineares. O fato de que o compartilhamento de informações entre indivíduos oferece uma vantagem

Algoritmo 1 Algoritmo ACO_ℝ (Socha, 2004; Socha & Dorigo, 2008).

$\tau \leftarrow$ distribuição de probabilidade inicial $P^i(x^i), i \in \{1..n\}$
enquanto Critério de parada não for satisfeito **faça**
 para cada formiga l **faça**
 $s_l^0 \leftarrow \emptyset$
 para cada componente i da solução **faça**
 escolher valor x^i aleatoriamente de acordo com distribuição de probabilidade
 $P^i(x^i)$
 $s_l^i \leftarrow s_l^{i-1} \cup x^i$
 fim para
 fim para
 $SI_{best} \leftarrow$ melhor solução da iteração
 $SG_{best} \leftarrow$ melhor solução entre as SG_{best} anteriores e a SI_{best}
 $\tau \leftarrow$ atualização do feromônio baseado em uma ou mais soluções encontradas
fim enquanto

evolutiva foi essencial no seu desenvolvimento. A PSO é baseada no comportamento social de pássaros, peixes e principalmente no comportamento social humano.

As partículas se movimentam por um espaço de busca e são capazes de armazenar informações passadas e compartilhar informações com outras partículas. Esses dois tipos de informações correspondem à aprendizagem individual (cognitiva) e à transmissão cultural (social). Dessa forma, as partículas utilizam as melhores soluções no seu processo de “evolução”. Kennedy & Eberhart (2001) usaram três princípios para explicar de forma breve o processo de adaptação cultural:

avaliar: cada partícula deve avaliar a solução encontrada por ela no espaço de busca;

comparar: cada partícula deve comparar a solução obtida por ela com as soluções obtidas pelas demais partículas;

imitar: as partículas devem imitar a partícula que mais se aproximou da solução desejada.

O compartilhamento de informações é realizado entre partículas consideradas vizinhas. A vizinhança considerada pelo PSO é topológica e não se refere ao espaço de atributos de cada partícula. Um possível arranjo topológico para considerar vizinhança entre as partículas está ilustrado na Figura 3.2. Há basicamente duas formas de se definir vizinhança nesse contexto, global e local. A vizinhança global considera que cada partícula é conectada a todas as outras e por isso o compartilhamento de informações ocorre entre todo o enxame. Na vizinhança local cada partícula é composta por ela própria e seus k -vizinhos mais próximos (Castro, 2006). Na Figura 3.3 é apresentada uma topologia para vizinhança global (à esquerda) e para vizinhança local com $k = 2$ (à direita).

A PSO pode ser aplicada basicamente a problemas binários ou contínuos. Cada partícula é representada pela sua posição atual, velocidade e melhor posição encontrada. Cada

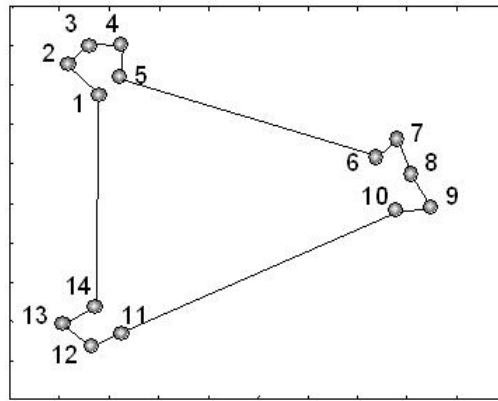


Figura 3.2: Representação da vizinhança topológica para partículas. (Fonte: <http://lsin.unisantos.br/lvcon/lvcon>).

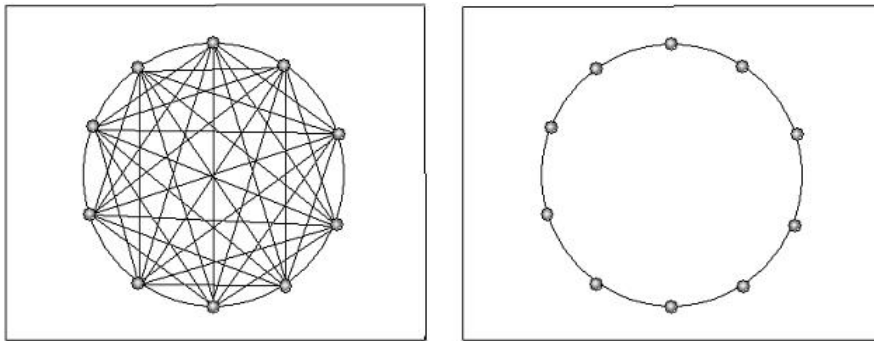


Figura 3.3: Representação da vizinhança global (à esquerda) e local (à direita). (Fonte: <http://lsin.unisantos.br/lvcon/lvcon>).

partícula é tratada como um ponto em um espaço D -dimensional. A posição da partícula i é dada por $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$; sua velocidade é dada por $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ e a melhor posição encontrada por esta partícula por $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ (Castro, 2006). Para o caso da vizinhança global, a melhor posição encontrada entre todas as partículas é representada pelo símbolo p_g .

Uma partícula irá se mover em uma determinada direção em função da sua posição atual, da sua velocidade, da melhor posição encontrada por ela e da melhor posição encontrada pelos seus vizinhos. As equações 3.1 e 3.2 determinam a forma com que a velocidade e a posição das partículas são atualizadas, respectivamente. Para evitar a explosão de velocidade das partículas um limite de velocidade foi introduzido. A posição da partícula pode também ser restrita ao intervalo do espaço de busca definido. No Algoritmo 2 são descritos os passos para o PSO.

$$v_{id}(t+1) = w \cdot v_{id}(t) + \varphi_1 \cdot r_1 \cdot (p_{id} - x_{id}(t)) + \varphi_2 \cdot r_2 \cdot (p_{gd} - x_{id}(t)) \quad (3.1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \quad (3.2)$$

onde t representa a iteração, w o peso da inércia, cujo papel é balancear a busca global e a local e foi introduzido por Shi & Eberhart (1998), r_1 e r_2 são dois valores independentes distribuídos uniformemente no intervalo $[0, 1]$, φ_1 e φ_2 são constantes de aceleração.

Algoritmo 2 Algoritmo PSO.

```

 $\tau \leftarrow 1$ 
enquanto Critério de parada não for satisfeito faça
  para cada partícula  $i$  faça
    se aptidão de  $x_{id} > p_{id}$  então
       $p_{id} \leftarrow x_{id}$ 
    fim se
    Atualizar velocidade da partícula conforme a Equação 3.1
    Atualizar posição da partícula conforme a Equação 3.2
  fim para
   $\tau \leftarrow t + 1$ 
fim enquanto

```

3.2 Sistema Imunológico Artificial

O Sistema Imunológico (SI) dos vertebrados é tão complexo quanto o cérebro. O funcionamento do SI é interessante não apenas do ponto de vista biológico, mas também sobre uma perspectiva computacional. O SI é responsável por inspecionar o organismo em busca de células do próprio corpo que apresentem mal funcionamento, como câncer e células tumorais, além de doenças causadas por elementos oportunistas. Todo elemento reconhecido pelo SI é chamado de antígeno (Ag).

O SIA é um sistema computacional inspirado na teoria da imunologia e funções imunológicas observadas e engloba princípios e modelos que são aplicados para solucionar problemas reais. SIAs têm sido aplicados em uma grande variedade de áreas de domínio, como reconhecimento de padrões e classificação, otimização, análise de dados e segurança de computadores.

Os principais aspectos relacionados ao desenvolvimento de um modelo baseado em SIAs são: uma representação para criar modelos abstratos de órgãos, células e moléculas imunológicas, um conjunto de funções de afinidade a fim de quantificar as interações entre os “elementos artificiais” e um conjunto de algoritmos de propósito geral para governar a dinâmica do SIA (Castro & Timmis, 2002).

No SI, o linfócito é a célula responsável pela imunidade. Existem dois tipos de linfócitos: os linfócitos T (células-T) e os linfócitos B (células-B). Apesar da grande similaridade, estas células diferem na forma de reconhecimento dos antígenos (Ags) e de sua função. O reconhecimento antigênico é a primeira etapa para que o SI inicie uma resposta imunológica. Os Ags que pertencem originalmente ao organismo, e são inofensivos ao seu

funcionamento, são denominados de antígenos próprios. Os antígenos que causam doenças ao organismo são chamados de antígenos não-próprios (Castro, 2006).

As células-T são maturadas no timo e protegidas por uma barreira de sangue capaz de excluir de forma eficiente antígenos não-próprios do timo, ou seja, aqueles que não pertencem originalmente ao ambiente tímico. As células-T que contêm receptores capazes de reconhecer Ags-próprios presentes no timo são eliminadas desse conjunto por meio de um processo chamado de seleção negativa. Todas as células-T que saem do timo para circular por todo o corpo são ditas tolerantes a elas próprias, ou seja, elas não são auto-reativas (Castro & Timmis, 2002; Abbas et al., 2000).

Com um papel complementar à seleção negativa, a seleção clonal é uma teoria que aborda a organização da resposta imunológica ao reconhecer, por meio das células-B, um padrão antigênico não-próprio. De forma breve, quando um antígeno não-próprio é reconhecido pelos receptores das células-B com uma certa afinidade, ele é selecionado para se proliferar e produzir anticorpos em grande volume. Os anticorpos são liberados pelas células-B para atuar contra os antígenos não-próprios invasores (Castro & Timmis, 2002; Abbas et al., 2000).

Como forma de modelar esse reconhecimento realizado pelos receptores das células, Perelson & Oster (1979) introduziram o conceito de forma-espço. Esse conceito diz que todas as propriedades dos receptores celulares que determinaram as interações entre as moléculas podem ser descritas por uma estrutura de dados. Assim um anticorpo Ab pode ser representado por um conjunto de coordenadas $Ab = \langle Ab_1, Ab_2, \dots, Ab_L \rangle$, e um antígeno por $Ag = \langle Ag_1, Ag_2, \dots, Ag_L \rangle$. Essa estrutura pode ser formada, por exemplo, por valores binários, reais ou inteiros, e é suficiente para quantificar a afinidade ou complementariedade entre moléculas. Para quantificar a afinidade no caso de anticorpos e antígenos formados por coordenadas cujos valores são reais pode se utilizar a distância Euclidiana ou de Manhattan, por exemplo. Para coordenadas binárias, pode-se utilizar a distância de Hamming, por exemplo (Castro, 2006).

Para problemas de otimização, ao invés de se utilizar antígenos usa-se uma função objetivo para avaliar a qualidade da resposta de cada anticorpo. Inspirados neste princípio Castro & Von-Zuben (2002) desenvolveram um algoritmo denominado CLONALG.

O algoritmo inicialmente gera uma população de anticorpos que em seguida são avaliados por uma função objetivo. Todos os anticorpos são clonados de acordo com suas aptidões. Quanto maior a aptidão, maior o número de cópias geradas. Cada clone produzido é mutado proporcionalmente ao seu valor de aptidão. Quanto maior a aptidão, menor a taxa de mutação. Finalmente um determinado número de anticorpos com as menores aptidões na população são substituídos pelos melhores indivíduos mutados. Esses passos são repetidos até um critério de parada ser satisfeito. No Algoritmo 3 é descrito o CLONALG para problemas de otimização, conforme (Castro, 2006).

Algoritmo 3 Algoritmo CLONALG para problemas de otimização.

Criar uma população inicial de anticorpos P

$t \leftarrow 1$;

enquanto Critério de parada não for satisfeito **faça**

 Avaliar aptidão de P

$P1 \leftarrow$ Selecionar $n1$ anticorpos de P com maiores aptidões

$C \leftarrow$ Clonar cada anticorpo de $P1$ proporcional a sua aptidão

$C1 \leftarrow$ Mutar cada anticorpo de C proporcional a sua aptidão

 Avaliar aptidão de $C1$

$P1 \leftarrow$ Selecionar $n1$ melhores clones mutados

$P \leftarrow$ Substituir os $n2$ anticorpos de menores aptidões de P pelos $n2$ de maior aptidão de $P1$

$t \leftarrow t + 1$

fim enquanto

3.3 Algoritmos Genéticos

A Computação Evolutiva (CE) é um paradigma de computação inspirado na teoria da seleção natural dos seres vivos, relatada por Charles Darwin em seu livro “A Origem das Espécies”. Segundo essa teoria, os indivíduos mais resistentes e bem adaptados possuem maior chance de sobrevivência, o que pode ser visto como um processo de “otimização natural”. A CE é aplicada principalmente na resolução de problemas de otimização e os algoritmos desse paradigma apresentam características como auto-organização e comportamento adaptativo (Goldberg & Holland, 1988). Segundo Carvalho et al. (2004), uma das principais características da CE é que seus algoritmos são constituídos por passos genéricos e adaptáveis, podendo ser utilizados na solução de diferentes problemas. Entre as áreas que fazem parte da CE, estão os AGs, propostos por Holland (1975). Porém, seu desenvolvimento ocorreu com maior intensidade a partir do trabalho de Goldberg & Holland (1988), no final da década de 80.

Em AGs, cada possível solução para um determinado problema é chamado de indivíduo ou cromossomo e o conjunto desses cromossomos forma uma população. Uma codificação desses indivíduos geralmente é necessária, sendo a binária a mais utilizada. Nesse caso, cada *bit* de um indivíduo é chamado de gene. Uma representação de um cromossomo binário formado por 10 genes é apresentado na Figura 3.4.

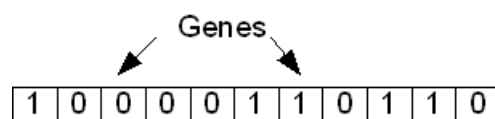


Figura 3.4: Representação de um cromossomo binário em um AG.

A busca por uma solução em AGs ocorre em um processo iterativo, em que cada iteração é denominada geração. A cada geração, o princípio da seleção é aplicado para

escolher os indivíduos que serão recombinados (*crossover*) e/ou modificados (mutação). Além disso, alguns algoritmos utilizam o conceito de elitismo, que seleciona indivíduos com os melhores valores de aptidão para a geração seguinte.

Um AG pode utilizar uma população inicial gerada aleatoriamente quando não há um conhecimento prévio sobre regiões mais promissoras dentro do espaço de busca. Cada indivíduo gerado é uma possível solução do problema em questão, que é avaliado por uma função de aptidão, que fornece uma medida de sua qualidade. Essa medida é utilizada durante a seleção de indivíduos, que pode ser realizada de diferentes formas.

O *crossover* é o operador genético responsável pelo cruzamento de características entre pares de indivíduos (pais) com o objetivo de gerar descendentes. Se o cruzamento de um ponto for utilizado, determina-se um ponto em que os cromossomos pais são divididos e cada descendente gerado será formado por exatamente um trecho de cada um dos cromossomos progenitores. Na Figura 3.5 é apresentada a aplicação do operador de *crossover* de um ponto. A frequência com que esse operador é aplicado é definido por uma taxa de cruzamento. O operador genético de mutação é responsável pela manutenção da diversidade genética na população, modificando um ou mais genes de um cromossomo. A taxa de mutação define a frequência de aplicação do operador de mutação que, geralmente, é aplicado com uma probabilidade baixa (Carvalho et al., 2003). A utilização do operador de mutação para um cromossomo binário é mostrado na Figura 3.6. No Algoritmo 4 são descritos os principais passos dos AGs, conforme (Carvalho et al., 2003).

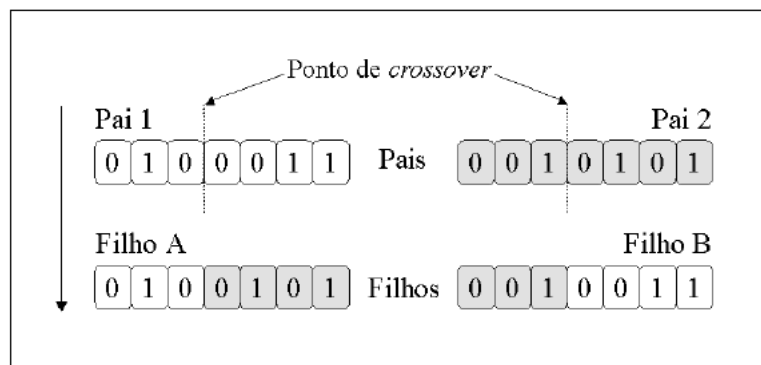


Figura 3.5: Operador genético de *crossover* de um ponto (Souto et al., 2003).

Para problemas de otimização de parâmetros em domínio contínuo, é mais natural representar os genes por números reais. O uso do alfabeto binário para representar valores no espaço contínuo pode produzir o efeito denominado penhasco de Hamming (*Hamming cliff*). Isso acontece quando a codificação binária de dois valores adjacentes difere em todos os bits. Por exemplo, os valores 31 e 32 são representados por 011111 e 100000, respectivamente (Herrera et al., 1998).

Com a utilização da codificação real para problemas de domínio contínuo, não há diferença entre a codificação e o espaço de busca (Herrera et al., 1998), pois cada gene

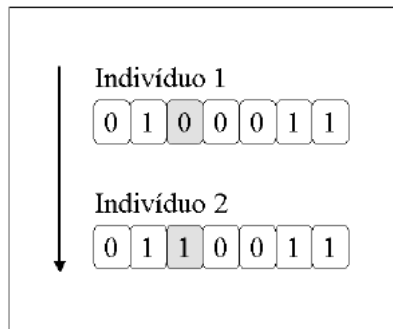


Figura 3.6: Operador genético de mutação (Souto et al., 2003).

Algoritmo 4 Algoritmo Genético.

$t = 0$;
 Gerar População Inicial $P(0)$
para cada indivíduo i da população atual $P(t)$ **faça**
 Avaliar aptidão do indivíduo i
fim para
enquanto Critério de parada não for satisfeito **faça**
 $t = t + 1$
 Selecionar população $P(t)$ a partir de $P(t - 1)$
 Aplicar operadores de cruzamento sobre $P(t)$
 Aplicar operadores de mutação sobre $P(t)$
 Avaliar $P(t)$
fim enquanto

representa uma variável do problema. Dessa forma, o tamanho do cromossomo tem o mesmo tamanho do vetor de soluções para o problema. Para empregar um AG com codificação real (AG_{CR}) na solução de problemas, foram desenvolvidos diferentes operadores de *crossover* e mutação. Em Herrera et al. (1998) são analisados diferentes operadores e os autores concluem que o operador de mutação não-uniforme (Michalewicz, 1992) e os operadores de *crossover* BLX- α (Eshelman & Schaffer, 1993) e *logical* FCB (Herrera et al., 1994) são os mais adequados para serem utilizados com o AG_{CR} . A seguir, o operador de *crossover* BLX- α e o operador de mutação não-uniforme são apresentados. Optou-se aqui por apresentar o BLX- α pela sua simplicidade se comparado ao *logical* FCB.

Sejam $C_1 = (c_1^1, \dots, c_1^n)$ e $C_2 = (c_1^2, \dots, c_1^n)$ dois cromossomos selecionados para aplicação do operador de *crossover*. Para o operador BLX- α , um descendente é gerado: $H = (h_1, \dots, h_i, \dots, h_n)$, onde h_i é um número aleatoriamente (uniformemente) escolhido no intervalo $[c_{min} - I \cdot \alpha, c_{max} + I \cdot \alpha]$, $c_{min} = \text{minimo}(c_i^1, c_i^2)$, $c_{max} = \text{maximo}(c_i^1, c_i^2)$, $I = c_{max} - c_{min}$ e α determina o balanceamento entre prospecção (*exploitation*) e exploração (*exploration*), e seu valor deve ser definido pelo usuário. No trabalho de Herrera et al. (1998), diferentes valores para α foram testados. O melhor balanceamento foi alcançado quando $\alpha = 0.5$. A seguir será explicado o operador de mutação não-uniforme.

Seja $C = (c_1, \dots, c_i, \dots, c_n)$ um cromossomo e $c_i \in [a_i, b_i]$ um gene que foi selecionado

aleatoriamente para mutação. A aplicação da mutação não-uniforme sobre esse gene resultará em um gene c'_i , cujo valor é definido da seguinte forma. Seja t a geração em que o operador está sendo aplicado e t_{max} o número máximo de gerações. Então

$$c'_i = \begin{cases} c_i + \Delta(t, b_i - c_i) & \text{se } \tau = 0 \\ c_i - \Delta(t, c_i - a_i) & \text{se } \tau = 1 \end{cases}$$

em que τ é um número aleatório binário (0 ou 1) e

$$\Delta(t, y) = y \left(1 - r^{\left(1 - \frac{t}{t_{max}}\right)^b} \right),$$

sendo r um número aleatório pertencente ao intervalo $[0, 1]$ e b um parâmetro escolhido pelo usuário que determina o grau de dependência do número de gerações. Em Herrera et al. (1998), foi utilizado $b = 5$.

3.4 Considerações Finais

Este capítulo iniciou com a apresentação das três frentes de estudo que a Computação Natural engloba. Posteriormente, os conceitos envolvidos na inteligência de enxames foram brevemente descritos e as metaheurísticas bioinspiradas ACO e PSO, pertencentes a esse paradigma foram abordadas.

Os SIAs também foram apresentados, com particular atenção ao princípio de seleção clonal. O funcionamento do algoritmo CLONALG para otimização, desenvolvido com base nesse princípio, foi explicado e seu algoritmo descrito. A última metaheurística abordada neste capítulo foram os AGs. Foram apresentados operadores de *crossover* e de mutação que podem ser utilizados com AGs com codificação real.

As técnicas bioinspiradas apresentadas neste capítulo foram utilizadas para ajustar os valores dos parâmetros livres das técnicas de AM descritas no Capítulo 2.

Capítulo 4

Experimentos com Funções de *Benchmark*

Algoritmos bioinspirados têm sido utilizados em diversas áreas, como robótica (Nolfi & Floreano, 2004), mineração de dados (Freitas, 2003; Sousa et al., 2004), otimização (Kennedy, 2005; Socha & Dorigo, 2008), segurança de redes (Castro & Von-Zuben, 2002), além de muitas outras. Neste estudo, foram investigados os algoritmos bioinspirados ACO, AG, CSA e PSO para o ajuste de parâmetros de técnicas de AM. Para testar a funcionalidade desses algoritmos, eles foram aplicados a cinco problemas de otimização bem conhecidos. Na literatura podem ser encontrados outros trabalhos que compararam algoritmos bioinspirados aplicados a problemas de otimização, como os dois trabalhos mencionados a seguir.

O trabalho de Elbeltagi et al. (2005) compara cinco algoritmos evolutivos. Nessa comparação, os autores utilizaram ACO, Algoritmos Meméticos (MA, do inglês, *Memetic Algorithms*) (Moscato, 1989), PSO, AG e *Shuffled Frog Leaping* (SFL) (Eusuff & Lansey, 2003) aplicados a dois problemas contínuos e um problema discreto. Segundo os autores, o PSO foi o que, geralmente, obteve melhores resultados em termos de taxa de sucesso e qualidade de solução, e o segundo melhor em termos de tempo de processamento.

Em Mezura-Montes & Lopez-Ramirez (2007), os autores compararam quatro algoritmos bioinspirados aplicados a quatro problemas de otimização com restrições. O objetivo era associar características das funções testadas com os algoritmos utilizados, a saber: Evolução Diferencial (ED) (Price et al., 2005), PSO, AG e $(\mu + \lambda)$ -Estratégia Evolutiva (EE) (Rechenberg, 1973). Os autores concluíram que o ED foi o mais competitivo para o conjunto de funções testadas, e o PSO o que convergiu mais rapidamente para regiões promissoras do espaço de busca.

Neste capítulo são apresentados os experimentos realizados com cinco funções de *benchmark*. Esses experimentos têm como objetivo avaliar a funcionalidade e o desempenho dos algoritmos bioinspirados investigados para funções unimodais e multimodais. Inicialmente, na Seção 4.1, são apresentadas as funções de *benchmark* utilizadas nos experi-

mentos preliminares. Os algoritmos bioinspirados implementados e seus parâmetros são apresentados na Seção 4.2. Na Seção 4.3 são apresentados os resultados obtidos. Finalizando, na Seção 4.4 são feitas algumas considerações finais sobre este capítulo.

4.1 Funções de *Benchmark*

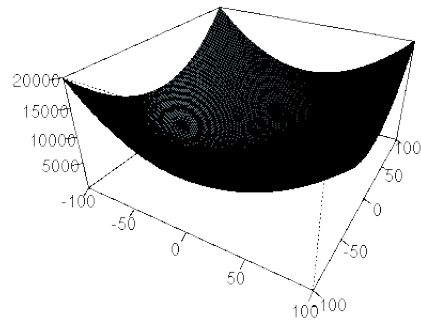
Na Tabela 4.1 são apresentadas as funções testadas, suas fórmulas, dimensões e intervalos de busca utilizados. Essas funções são comumente utilizadas para comparar algoritmos evolutivos (Bäck, 1996). Na Figura 4.1 estão ilustrados os gráficos bidimensionais dessas funções¹. As dimensões e os intervalos utilizados para cada função foram os mesmos utilizados por Elshamy et al. (2007). Os autores desse trabalho realizaram experimentos com modificações propostas por eles para o PSO e compararam com o PSO modificado, proposto por Shi & Eberhart (1998), utilizando vizinhança local e vizinhança global.

Tabela 4.1: Funções de *benchmark* utilizadas para verificação da funcionalidade e desempenho dos algoritmos bioinspirados investigados.

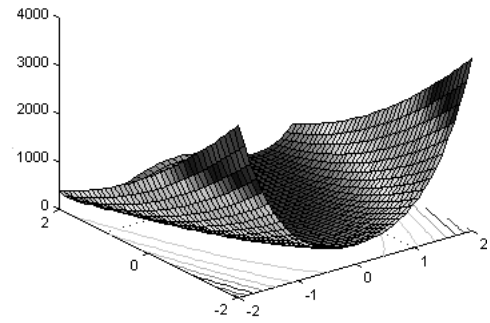
Função	Equação	Dim.	Intervalo
Esfera	$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$
Rastrigin	$f_3(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$
Schaffer f6	$f_4(x) = 0.5 + \frac{\text{sen}^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001(x^2+y^2))^2}$	2	$[-100, 100]$
Ackley	$f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$

As funções Esfera e Rosenbrock (Rosenbrock, 1960) são funções unimodais e testam a habilidade do algoritmo frente à superfícies planas. As funções Rastrigin (Aimo & Zilinskas, 1989), Schaffer f6 (Schaffer et al., 1989), e Ackley (Ackley, 1987) são multimodais e com um grande número de mínimos locais, testando assim a habilidade dos algoritmos em escapar de mínimos locais (Elshamy et al., 2007). O mínimo global para todas as funções é zero. Para isso, todas as variáveis x_i precisam ser iguais a zero, com exceção da função Rosenbrock, em que todas as variáveis x_i precisam ser iguais a um.

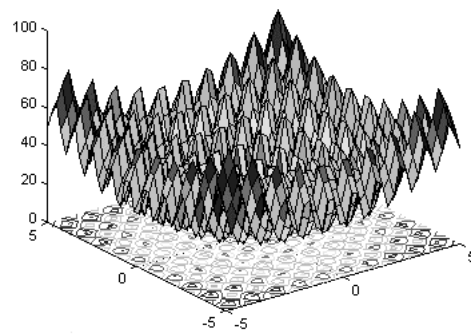
¹As figuras 4.1(a), 4.1(b), 4.1(c) e 4.1(e) foram obtidas do site: <http://www.geatbx.com/docu/fcnindex-01.html>



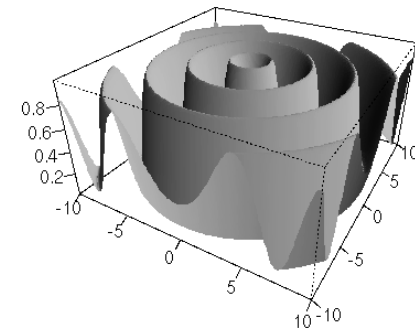
(a) Gráfico da função Esfera



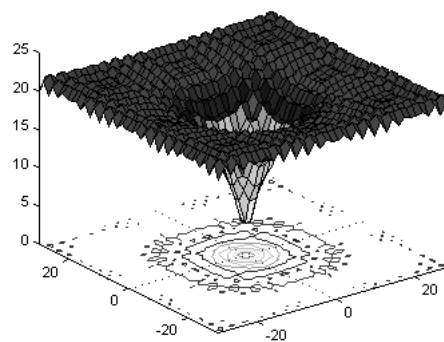
(b) Gráfico da função Rosenbrock



(c) Gráfico da função Rastrigin



(d) Gráfico da função Schaffer f6



(e) Gráfico da função Ackley

Figura 4.1: Gráfico das funções de *benchmark* utilizadas.

4.2 Algoritmos Bioinspirados Implementados

Quatro algoritmos bioinspirados foram implementados para a realização dos experimentos: Otimização por Enxame de Partículas (PSO, do inglês, *Particle Swarm Optimization*), Algoritmo de Seleção Clonal (CSA, do inglês, *Clonal Selection Algorithm*), Otimização por Colônia de Formigas (ACO, do inglês, *Ant Colony Optimization*) e Algoritmos Genéticos (AGs). Os algoritmos foram implementados na linguagem R. O R é um *software* livre para computação estatística e possui uma grande quantidade de pacotes disponíveis gratuitamente, muitos deles podem ser obtidos pelo site do projeto R².

O algoritmo ACO para domínios contínuos foi implementado com base em Socha (2004) e Socha & Dorigo (2008). A principal diferença para o ACO clássico é o uso de uma função de probabilidade contínua, ao invés de uma função de probabilidade discreta. Os parâmetros do ACO foram definidos com base no estudo de Socha & Dorigo (2008).

Um AG com codificação real foi implementado usando seleção por torneio, operador de *crossover* BLX- α e operador de mutação não-uniforme. Em razão dos resultados obtidos durante alguns experimentos realizados com funções de *benchmark*, o elitismo não foi usado. Para funções multimodais é importante explorar várias regiões do espaço de busca para não ficar preso à mínimos locais. Portanto, melhores resultados foram obtidos sem o uso do elitismo, que aumenta a prospecção em detrimento da exploração. Os valores usados para os parâmetros do AG são baseados em valores comumente usados na literatura, como em Huang & Wang (2006) e Elbeltagi et al. (2005).

O algoritmo PSO foi implementado como especificado em Shi & Eberhart (1998), em que um peso de inércia é adicionado ao algoritmo PSO original (Kennedy & Eberhart, 1995). A topologia de vizinhança global foi adotada. Os valores usados para aceleração, peso de inércia e velocidade das partículas foram definidos conforme Eberhart & Shi (2000).

Os anticorpos do algoritmo CSA foram desenvolvidos com codificação real. O número de anticorpos selecionados para clonagem foi definido de acordo com Castro & Von-Zuben (2002). O procedimento que substitui anticorpos da memória não foi usado. Ao invés disso, foi aplicado o operador de idade (Cutello et al., 2006; Castrogiovanni et al., 2007). Esse operador elimina anticorpos que não mutaram com sucesso por um determinado número de iterações. O número de clones gerados é proporcional à solução da qualidade gerada pelo anticorpo. A hipermutação é inversamente proporcional à qualidade do anticorpo.

Na Tabela 4.2 são descritos os valores usados para os parâmetros dos algoritmos bioinspirados aplicados para otimizar as funções de *benchmark*.

Nos experimentos com funções de *benchmark*, cada indivíduo dos algoritmos bioinspirados é composto por V_t números reais, em que V_t é o número de variáveis da função

²<http://www.r-project.org/>

Tabela 4.2: Valores para os parâmetros dos algoritmos bioinspirados.

Alg.	Parâmetro	Símbolo	Valor
ACO	Número de formigas	P_{an}	60
	Velocidade de convergência	ξ	0.70
	Local do processo de busca	q	10^{-4}
	Tamanho do arquivo de soluções	k	60
PSO	Número de partículas	P_{pa}	60
	Constante de aceleração 1	φ_1	1.494
	Constante de aceleração 2	φ_2	1.494
	Peso de controle da força de inércia	w	0.729
	Velocidade máxima	v	0.7
CSA	Número de anticorpos	P_{ab}	40
	Taxa da população selecionada para clonagem	n_1	1
	Tempo de vida	lt	6
	Taxa de hipermutação	h_r	0.5
AG	Número de cromossomos	P_{ch}	60
	Probabilidade de crossover	p_c	0.7
	Probabilidade de mutação	p_m	0.05
	Pressão seletiva	p_s	3
	Balanço entre exploração e prospecção	α	0.5
	Grau de dependência do número de iterações	b	5

a ser minimizada. Nos experimentos para ajuste de parâmetros, cada indivíduo é composto por N_t números reais, em que N_t é o número de parâmetros a serem ajustados para a técnica de classificação. As soluções iniciais (valores para os parâmetros das técnicas de classificação e valores para as variáveis das funções) são geradas aleatoriamente sobre os intervalos definidos. Como os algoritmos bioinspirados empregados são estocásticos, foram executadas 30 repetições para cada conjunto de dados ou funções, com diferentes populações iniciais. O valor 30 foi definido por ser a quantidade mínima de dados em uma população com distribuição normal, conforme o teorema central do limite (Achcar & Rodrigues, 2000).

A minimização das funções de *benchmark* é realizada até que uma condição de parada para os algoritmos bioinspirados seja satisfeita. Essa condição pode ser um número máximo de iterações ou um erro mínimo a ser alcançado, por exemplo. Em todos os experimentos realizados neste trabalho, o número máximo de iterações foi estabelecido como critério de parada. Esse critério foi adotado porque possibilita verificar a convergência de todos os algoritmos bioinspirados para o mesmo número de iterações.

4.3 Resultados

O número de iterações adotado para todos os algoritmos bioinspirados é o mesmo utilizado por Elshamy et al. (2007), ou seja, 10 000. Ao final das iterações, o algoritmo retorna o menor valor encontrado para a função. Como os algoritmos são estocásticos, foram realizadas 30 execuções independentes para cada função.

Na Tabela 4.3 são apresentadas as médias e os desvios padrão (em parênteses) das 30 execuções dos algoritmos bioinspirados para as funções unimodais. O menor valor para cada função está destacado. A convergência dos algoritmos bioinspirados para essas funções pode ser analisada por meio dos gráficos exibidos na Figura 4.2.

Em relação a função Esfera, pode-se observar que o PSO conseguiu valores muito menores que os demais algoritmos implementados. Em razão dos resultados publicados em Elshamy et al. (2007), era esperado que o PSO conseguisse valores próximos do mínimo global. Para a função Rosenbrock, o PSO novamente se sobressaiu aos demais algoritmos. Dos algoritmos implementados, o AG foi o segundo melhor para a função Esfera e o pior para a função Rosenbrock.

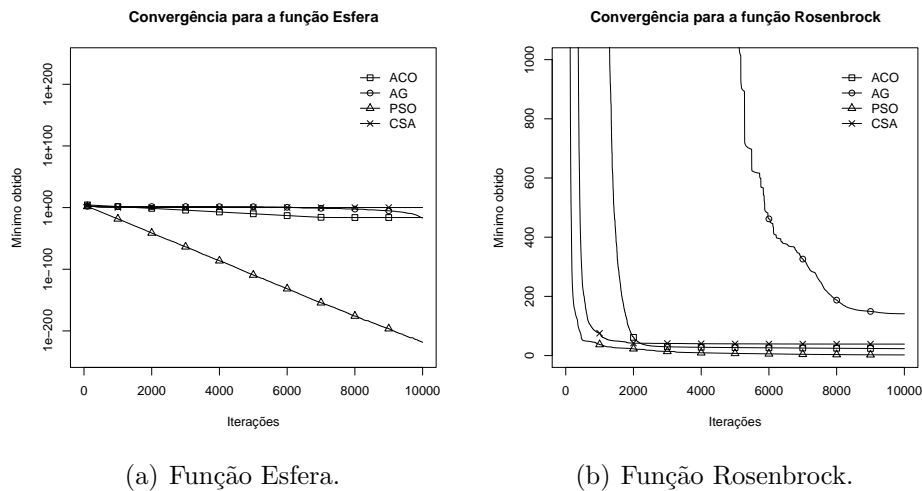
Para a função Esfera, é possível observar a partir do gráfico da Figura 4.2(a), que o PSO não ficou preso a um mínimo local durante as 10 000 iterações realizadas. O mesmo não ocorreu para o algoritmo AG e o CSA, que ficaram presos a um mínimo local logo nas primeiras iterações, e assim não conseguiram resultados muito próximos do mínimo global. O AG conseguiu nas últimas iterações reduzir um pouco o mínimo local encontrado. A convergência do ACO foi muito lenta, e após a iteração 7 000, aproximadamente, estagnou.

Por meio dos gráficos de convergência para a função Rosenbrock, que são apresentados no gráfico da Figura 4.2(b), pode-se observar que o CSA e o PSO convergiram rapidamente para valores próximos do mínimo global. O ACO convergiu um pouco mais lentamente, chegando a regiões promissoras do espaço de busca encontradas pelo CSA e PSO por volta da iteração 2 000. O AG apresentou os piores resultados para os dois critérios avaliados para esse problema. Convergiu mais lentamente para um mínimo local e obteve o valor mais distante do mínimo global.

Com os resultados apresentados para as funções unimodais, pode-se afirmar que o PSO superou os demais algoritmos testados para os dois critérios analisados. Convergiu rapidamente para áreas promissoras e conseguiu os menores resultados para as funções.

Tabela 4.3: Resultados para cada algoritmo bioinspirado para as funções unimodais após 10 000 iterações.

Algoritmo	Esfera	Rosenbrock
ACO	7.8e-17 (5.2e-17)	2.3e+01 (1.2e+01)
AG	1.4e-17 (2.6e-17)	1.4e+02 (1.3e+02)
CSA	1.3e+00 (1.2e-01)	3.9e+01 (1.3e+00)
PSO	4.0e-219 (0.0e+00)	2.2e+00 (2.3e+00)
Elshamy et al. (2007)	1.1e-152	6.9e+000



(a) Função Esfera.

(b) Função Rosenbrock.

Figura 4.2: Convergência dos algoritmos bioinspirados para funções unimodais.

Na Tabela 4.4 são apresentadas as médias de 30 execuções dos valores mínimos obtidos pelos algoritmos bioinspirados para as funções multimodais. Nessa tabela também são apresentados os melhores resultados do trabalho de Elshamy et al. (2007). Os desvios padrão são apresentados entre parênteses, e os menores valores para cada função estão destacados. Na Figura 4.3 são apresentados os gráficos de convergência para essas funções.

Todos os algoritmos bioinspirados tiveram grande dificuldade na minimização da função Rastrigin. Isso pode ser observado pela distância do mínimo global. Os algoritmos ficaram presos aos inúmeros pontos de mínimos locais existentes. É possível observar, por meio do gráfico ilustrado na Figura 4.3(a), que os algoritmos ACO e CSA não ficaram presos a um mínimo local. Porém, a convergência desses algoritmos foi muito lenta. O AG obteve o menor valor para a função Rastrigin e também convergiu tão rápido quanto os demais algoritmos. O PSO também convergiu rapidamente, mas ficou preso a mínimos locais e não superou o AG.

O gráfico dos resultados obtidos para a função Schaffer f6 é apresentado na figura 4.3(b). Como pode ser observado, os algoritmos bioinspirados ACO, AG e PSO convergiram rapidamente para um mínimo local. Porém, ficaram presos em mínimos locais e não conseguiram reduzir significativamente o valor encontrado após a iteração 2000, aproximadamente. A única exceção foi o CSA, que apresentou uma rápida convergência, sem ficar preso a mínimos locais. A partir da iteração 5000, aproximadamente, a velocidade de convergência reduziu, mas não estagnou, até o final das iterações. O AG, apesar do melhor resultado para a função Rastrigin, foi o pior algoritmo para a função Schaffer f6.

O gráfico apresentado na Figura 4.3(c) mostra que o ACO obteve o menor valor para a função Ackley. Esse algoritmo convergiu um pouco mais lentamente que os demais algoritmos. Porém, não ocorreu a convergência prematura para um mínimo local. Os demais algoritmos estagnaram logo nas primeiras iterações. Assim como para a função

Rastrigin, o CSA novamente não conseguiu bons resultados, e foi o pior algoritmo. O PSO e o AG conseguiram resultados similares ao ACO.

Não é possível apontar o melhor algoritmo para as funções unimodais. Para cada uma das três funções testadas, o valor mínimo foi obtido por três algoritmos diferentes. O PSO conseguiu o segundo menor valor para as três funções. Portanto, caso fosse necessário optar por um algoritmo, o PSO seria o mais adequado. Os resultados mostram que é importante analisar as características do problema antes de optar por um dos algoritmos analisados.

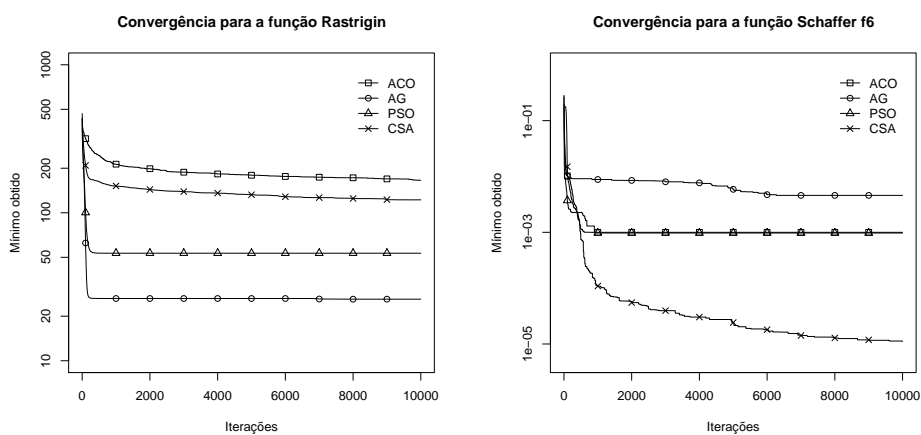
Tabela 4.4: Resultados para cada algoritmo bioinspirado para as funções multimodais após 10 000 iterações.

Algoritmo	Rastrigin	Schaffer f6	Ackley
ACO	1.7e+02 (1.5e+01)	1.0e-03 (1.1e-03)	1.0e-13 (4.8e-14)
AG	2.6e+01 (6.6e+00)	4.6e-03 (3.3e-03)	2.3e+00 (1.0e+00)
CSA	1.2e+02 (1.3e+01)	1.1e-05 (7.3e-06)	1.9e+01 (2.0e-01)
PSO	5.3e+01 (1.4e+01)	9.7e-04 (3.0e-03)	7.9e-01 (8.1e-01)
Elshamy et al. (2007)	3.4e+001	0	3.0e-002

Apesar da robustez dos algoritmos bioinspirados (Lourenço et al., 2002; Ide & Yasuda, 2005), seus desempenhos podem variar devido ao ajuste de seus parâmetros. Uma maior velocidade de convergência para o ACO, por exemplo, pode ser obtido com o aumento desse parâmetro (ξ). Porém, isso pode resultar em uma convergência prematura do algoritmo para um mínimo local.

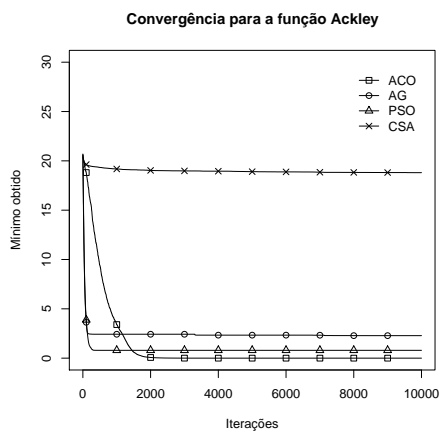
4.4 Considerações Finais

Os experimentos realizados analisaram dois aspectos dos algoritmos bioinspirados frente a funções unimodais e multimodais. Os aspectos considerados foram a velocidade de convergência e a proximidade dos valores encontrados em relação ao mínimo global. Em geral, os resultados mostram que o PSO se sobressaiu para funções unimodais, encontrando os menores valores para esses problemas com um número reduzido de iterações. Para as funções multimodais não foi possível determinar o melhor algoritmo. Entretanto, o PSO conseguiu o segundo menor valor para as três funções, e convergia rapidamente para o mínimo local. Para Schaffer f6, o CSA conseguiu os menores valores. Porém, sua convergência ocorreu lentamente. A principal contribuição destes experimentos foi verificar a funcionalidade dos algoritmos bioinspirados e relacioná-los com as características das funções unimodais e multimodais utilizados. Essa relação pode auxiliar na escolha algoritmo mais apropriado para um dado problema.



(a) Rastrigin.

(b) Schaffer f6.



(c) Ackley.

Figura 4.3: Convergência dos algoritmos bioinspirados para funções multimodais.

Capítulo 5

Experimentos de Ajuste de Parâmetros

Como mencionado anteriormente, este estudo tem como objetivo utilizar algoritmos de otimização bioinspirados para ajustar os valores dos parâmetros livres de técnicas de classificação. Esses algoritmos serão comparados para o ajuste de parâmetros utilizando como critério o desempenho obtido pelos classificadores para dados de expressão gênica. Essa escolha teve por motivação restringir o domínio e o formato dos dados de forma a investigar se um algoritmo é mais adequado para esse domínio. Para avaliar os algoritmos com outros formatos e domínios, foram usados também, em alguns experimentos, conjuntos de dados do repositório UCI (Asuncion & Newman, 2007).

Nesta seção, são apresentados os três métodos experimentais empregados para o ajuste de parâmetros das técnicas de classificação SVMs e RNs. Esses três métodos representam a evolução ao longo do mestrado na definição do método a ser empregado para obter estimativas de desempenho de classificação confiáveis e evitar o uso indireto dos dados do conjunto de teste no processo de busca dos algoritmos bioinspirados.

Este capítulo está organizado como segue. Na Seção 5.1, são descritas as principais características dos conjuntos de dados utilizados nos experimentos e as operações de pré-processamento realizadas. Na Seção 5.2 são apresentados os algoritmos bioinspirados que foram aplicados ao problema de ajuste de parâmetros. Na Seção 5.3 são apresentadas as ferramentas de *software* utilizadas que implementam as técnicas de classificação SVMs e RNs. As métricas e os métodos de avaliação dos classificadores são apresentados na Seção 5.4. No método experimental A, apresentado na Seção 5.5, são definidos os métodos e os programas empregados para o ajuste de parâmetros de SVMs. As estimativas de desempenho são obtidas por meio da validação cruzada estratificada com k partições. No método experimental B, apresentado na Seção 5.6, são definidos os métodos empregados para o ajuste de parâmetros de RNs. Os resultados obtidos são baseadas em dois laços aninhados com validação cruzada estratificada. O método experimental C, apresentado na Seção 5.7, é semelhante ao método experimental B. As principais diferenças estão

no processo de ajuste de parâmetros, em que os valores para os parâmetros são obtidos para cada partição do laço mais externo, e no armazenamento das estimativas. Este método experimental é usado para o ajuste de parâmetros de RNs e SVMs. Os resultados experimentais obtidos para cada método são apresentados no final da respectiva seção.

5.1 Conjuntos de Dados

Na Tabela 5.1 são descritos os conjuntos de dados utilizados nos experimentos relatados neste capítulo e suas principais características. Nessa tabela são apresentados, para cada conjunto de dados, o número de exemplos (# Ex.), o número de atributos (# Attr.) (desconsiderando o atributo classe), o número de classes do problema (# Classes), o número de exemplos por classe (# Ex. por classe) e a Taxa de Erro Minoritária (TEM). A TEM é a taxa de erro de um classificador que sempre prediz como saída a classe com o maior número de exemplos, não apresentando, portanto, capacidade de generalização.

Tabela 5.1: Conjuntos de dados utilizados nos experimentos.

Conjunto	# Ex.	# Attr.	# Classes	# Ex. por classe	TEM (%)
Cólon	62	32	2	40/22	35.5
Glioma	50	16	2	22/28	44.0
Leucemia	100	50	2	49/51	49.0
Pâncreas	49	50	2	26/23	46.9
Leucemia2	248	50	6	79/15/27/20/43/64	31.9
Pulmão	86	50	3	28/37/21	43.0
<i>Australian</i>	690	14	2	307/383	44.5
<i>Pima</i>	768	8	2	268/500	34.9
<i>Segment</i>	2310	19	7	330/330/330/330/330/330/330	14.3
<i>Vehicle</i>	846	18	4	218/212/217/199	23.5

Os conjuntos Cólon (Alon et al., 1999), Glioma (Nutt et al., 2003), Leucemia (Haslinger et al., 2004), Pâncreas (Ishikawa et al., 2005), Leucemia2 (Yeoh et al., 2002) e Pulmão (Beer et al., 2002) contêm dados de expressão gênica.

A análise de dados de expressão gênica pode ser realizada com diferentes propósitos, como classificar células entre normais e tumorais. O processo de expressão gênica consiste na conversão de instruções contidas em trechos da cadeia de DNA (do inglês, *Desoxyribonucleic Acid*), chamados de genes, em proteínas. A transcrição do DNA em RNA (*Ribonucleic Acid*) mensageiro (mRNA) e a tradução do mRNA em proteína, constituem o dogma central da Biologia Molecular. A quantidade de mRNA presente em uma célula e de proteína produzida estão relacionadas, porém nem sempre são iguais (De-Robertis & Hib, 2001; Faceli et al., 2005). Há duas abordagens para avaliar a expressão gênica: a análise do transcriptoma e a análise do proteoma. A primeira utiliza o produto da transcrição, ou seja, o mRNA. A segunda analisa a proteína diretamente. Apesar da análise do

proteoma ser feita diretamente com o produto final da expressão de um gene, essa análise é muito mais trabalhosa do que a do transcriptoma. Com o avanço de tecnologias ligadas à análise do proteoma, o uso dessa abordagem tem crescido. Entretanto, o transcriptoma é ainda a forma mais utilizada para análise de dados de expressão gênica (Faceli et al., 2005). A mensuração da expressão gênica por meio da análise do transcriptoma pode ser feita por diversas técnicas, que geram dados em larga escala, entre elas encontram-se os métodos *Serial Analysis of Gene Expression* (SAGE) (Velculescu et al., 1995) e *microarray* (Schena et al., 1995).

Os conjuntos de dados *australian* (*Australian Credit Approval*), *pima* (*Pima Indians Diabetes*), *segment* (*Image Segmentation*) e *vehicle* (*Vehicle Silhouettes*) foram usados no projeto *Statlog* (Michie et al., 1994) e foram obtidos do repositório UCI. A nomenclatura original dos conjuntos de dados foi mantida em inglês, pois eles são amplamente conhecidos por esses nomes. Todos os conjuntos de dados possuem apenas atributos numéricos, com exceção do conjunto *australian*, que possui 8 atributos categóricos.

Pré-processamento dos dados

Duas operações de processamento dos dados foram realizadas. Primeiro, os atributos de entrada foram normalizados para que ficassem com média zero e desvio padrão um, para os dados a serem usados pelas SVMs, e entre zero e um para as RNs. A normalização adotada para as SVMs é amplamente utilizada em experimentos de AM. Porém, para RNs, é comum a normalização entre zero e um, pois valores próximos de zero geram uma pequena correção nos pesos da rede. Para os experimentos reportados na Seção 5.7, os dados não foram normalizados para SVMs, pois esse procedimento é realizado internamente pela implementação dessa técnica. A normalização dos dados foi embutida na validação cruzada, ou seja, a média e o desvio padrão para a normalização foram definidos com base apenas nos dados de treinamento e posteriormente usados nos dados de teste.

A segunda operação realizada foi a seleção de atributos. A seleção de atributos em análise de dados de expressão gênica, chamada de seleção de genes, é importante não apenas para reduzir o custo computacional, pois normalmente há milhares de genes, mas também eliminar genes não informativos. A seleção de atributos dos conjuntos de dados Cólon e Glioma foram feitos por Souza (2005). O autor utilizou o método *Threshold Number of Misclassification* (TNoM) (Ben-Dor et al., 2000) com 32 e 16 atributos, respectivamente. Para os conjuntos de dados Pâncreas e Leucemia, o método *between-groups to within-groups sum of squares* (BSS/WSS) (Dudoit et al., 2002) foi usado com 50 atributos para ambos. O melhor número de genes varia entre os conjuntos de dados. O valor 50 foi escolhido por ser um número pequeno de genes, o que acelera o processo de classificação, e por ser um valor utilizado nos trabalhos de seleção de genes, como por exemplo

em Statnikov et al. (2005a). Para os experimentos reportados na Seção 5.7, o método BSS/WSS foi embutido no processo de validação cruzada, ou seja, a seleção de atributos foi realizada apenas usando o conjunto de treinamento,

Neste trabalho, não foi realizado nenhum estudo para determinar o melhor número de genes para cada conjunto de dados. Portanto, o problema de viés na seleção de genes é reconhecido. Porém, esse viés não anula o propósito deste estudo, o qual não é ter o melhor classificador possível, mas comparar diferentes algoritmos bioinspirados e abordagens para o ajuste de parâmetros das técnicas de classificação.

5.2 Algoritmos Bioinspirados

Os algoritmos bioinspirados usados durante os experimentos apresentados neste capítulo são os mesmos usados nos experimentos com funções de *benchmark* no Capítulo 4. As diferenças se resumem aos valores de parâmetros adotados para esses algoritmos. Para facilitar a visualização desses parâmetros, a Tabela 4.2, com os parâmetros desses algoritmos, foi usada como base para a construção da Tabela 5.2, com as devidas modificações nos valores dos parâmetros.

O processo de ajuste dos parâmetros das técnicas de classificação é realizado até que uma condição de parada para os algoritmos bioinspirados seja satisfeita. Essa condição pode ser um número máximo de iterações ou um erro mínimo a ser alcançado, por exemplo. Em todos os experimentos realizados neste trabalho, o número máximo de iterações foi estabelecido como critério de parada. A decisão por esse critério ocorreu por não ser possível definir um erro mínimo comum para todos os conjuntos de dados e funções testadas.

Este trabalho não compara o desempenho dos algoritmos bioinspirados em relação ao tempo de processamento. Esse fator é muito dependente da linguagem de programação, configuração do computador usado para rodar os experimentos e habilidades do programador. Ao invés disso, compara-se o número de execuções necessárias das funções de aptidão, que neste trabalho, são as técnicas de classificação. Embora esse método trate dos problemas associados com o tempo de processamento, isso não considera a complexidade de tempo dos algoritmos (Socha & Dorigo, 2008).

Os classificadores gerados com os conjuntos de treinamento são utilizados como função de aptidão. Os algoritmos bioinspirados usam a taxa de erro de validação gerada por esses classificadores para guiarem suas buscas, ou seja, o objetivo dos algoritmos é minimizar essa métrica, que é calculada para os diferentes métodos experimentais implementados. A taxa de erro de um classificador para um conjunto de dados é dada pela Equação 2.2.

Tabela 5.2: Valores para os parâmetros dos algoritmos bioinspirados.

Alg.	Parâmetro	Símbolo	Valor
ACO	Número de formigas	P_{an}	30
	Velocidade de convergência	ξ	0.70
	Local do processo de busca	q	10^{-4}
	Tamanho do arquivo de soluções	k	30
PSO	Número de partículas	P_{pa}	30
	Constante de aceleração 1	φ_1	1.494
	Constante de aceleração 2	φ_2	1.494
	Peso de controle da força de inércia	w	0.729
	Velocidade máxima	v	0.7
CSA	Número de anticorpos	P_{ab}	40
	Taxa da população selecionada para clonagem	n_1	1
	Tempo de vida	lt	6
	Taxa de hipermutação	h_r	0.5
AG	Número de cromossomos	P_{ch}	30
	Probabilidade de crossover	p_c	0.7
	Probabilidade de mutação	p_m	0.05
	Pressão seletiva	p_s	3
	Balanço entre exploração e prospecção	α	0.5
	Grau de dependência do número de iterações	b	5

5.3 Técnicas de Classificação Investigadas

Para a geração dos classificadores, foram utilizados pacotes do ambiente R que implementam os algoritmos de aprendizado para geração de modelos de SVMs e RNs para classificação.

Para as SVMs, foi utilizado o pacote *e1071*, que é uma interface para a biblioteca LIBSVM, desenvolvida por Meyer (2001). Essa biblioteca implementa diferentes *kernels* para as SVMs, e cada um deles possui diferentes parâmetros. O *kernel* RBF (Gaussiano) foi utilizado nesta pesquisa por ser o que apresenta, em geral, melhores resultados e menos dificuldades numéricas do que outros *kernels* (Hsu et al., 2007). Para o uso do *kernel* RBF é necessário ajustar dois parâmetros, um referente ao *kernel* RBF, denominado *gama* (γ) e o outro à própria SVM, denominado custo (C), conforme descrito na Seção 2.2.

Para as RNs, existem diferentes pacotes do ambiente R que poderiam ser utilizados, como o *AMORE*, o *neural* e o *nnet*. Neste trabalho, foi utilizado o pacote *AMORE* com uma rede Perceptron Multi-camadas (MLP, do inglês, *Multi-Layer Perceptron*) treinada pelo algoritmo *backpropagation* com *momentum*. O pacote *AMORE* foi escolhido porque propicia maior flexibilidade em relação aos parâmetros da rede, possibilitando um maior

controle sobre ela. Entre os parâmetros que esse pacote permite ajustar, estão a taxa de aprendizado da rede, o termo *momentum*, a função de ativação, o número de camadas e o número de neurônios em cada camada. A MLP é a arquitetura de rede mais popular, e tem sido empregada com sucesso em uma grande variedade de problemas. Redes MLP possuem uma ou mais camadas ocultas de neurônios, cujo pesos das conexões são normalmente ajustados por algoritmos de aprendizado baseados em gradiente. Nos experimentos realizados, foi usada apenas uma camada oculta, que possibilita a representação de inúmeras funções, suficientes para o propósito desse estudo, o qual não é ter a melhor arquitetura possível, mas comparar diferentes abordagens para o ajuste de parâmetros. As RNs MLP podem ser totalmente ou parcialmente conectadas. Uma rede neural é denominada totalmente conectada quando cada um dos nós de uma camada está conectado a todos os nós da camada seguinte e é dita parcialmente conectada quando alguns neurônios de camadas adjacentes não estiverem conectados. As RNs utilizadas nesta pesquisa são totalmente conectadas. Os parâmetros ajustados foram a taxa de aprendizado, o termo *momentum* e o número de neurônios na camada oculta.

5.4 Avaliação dos Experimentos

Para avaliar o ajuste de parâmetros realizado pelos algoritmos bioinspirados e outras abordagens, os classificadores gerados com os melhores parâmetros encontrados foram comparados em relação à taxa de erro de classificação obtida para diferentes conjuntos de dados. A taxa de erro foi utilizada pelos algoritmos bioinspirados para guiarem suas buscas pelo conjunto de valores para os parâmetros das técnicas de classificação. Portanto, essa métrica é utilizada na comparação dos resultados obtidos. Além disso, em problemas de classificação binária, os experimentos da Seção 5.7 também foram analisados por meio de gráficos de curva ROC (do inglês, *Receiver Operating Characteristics*) e área sob a curva ROC (AUC, do inglês, *Area Under an ROC Curve*).

Para avaliar se houve diferença estatística entre os diferentes classificadores gerados, foram aplicados testes estatísticos. Inicialmente, foi aplicado o teste *t* pareado corrigido, desenvolvido por Nadeau & Bengio (2003) e descrito na Seção 2.1.2. Com esse teste, apenas dois algoritmos podem ser comparados de cada vez. Para os experimentos das seções 5.6 e 5.7, também foi aplicado o teste de Friedman (Demšar, 2006), que permite a comparação de múltiplos algoritmos e múltiplos conjuntos de dados. Como apresentado na Seção 2.1.2, esse teste permite verificar se há ou não diferenças significativas entre os diferentes classificadores para todos os conjuntos de dados. Nos testes em que houveram diferenças significativas, o teste à *posteriori* Nemenyi (Demšar, 2006) foi aplicado para verificar onde ocorreram as diferenças.

5.5 Método Experimental A: Ajuste de Parâmetros de SVMs

Neste método experimental, a geração do classificador é baseado no método de reamostragem de validação cruzada estratificada com k partições. Uma, dentre as k partições, foi separada para teste e as demais foram usadas para treinamento. Do conjunto de treinamento, uma partição é separada para validação. Os exemplos presentes no conjunto de validação são usados para estimar o erro de generalização das SVMs. A taxa de erro médio obtido para as k partições de validação é a estimativa para a taxa de erro médio de teste do classificador. Essa estimativa é usada pelos algoritmos bioinspirados para guiarem suas buscas no espaço de soluções à procura do conjunto ótimo de valores para os parâmetros das SVMs. Para cada solução (combinação de valores dos parâmetros), o algoritmo de aprendizado é executado k vezes. Na Figura 5.1 é apresentado um exemplo desse método experimental para $k = 4$ partições.

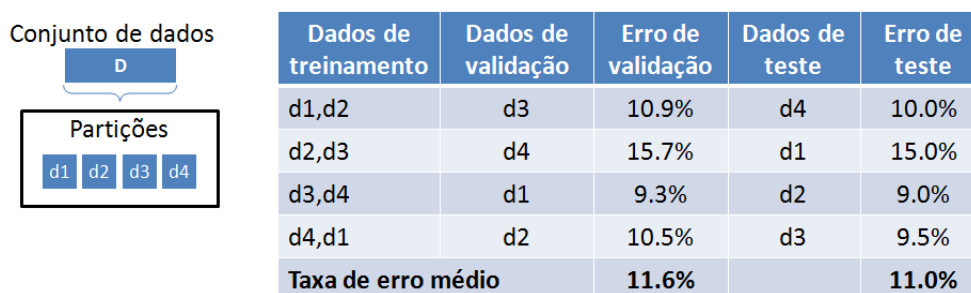


Figura 5.1: Método Experimental A.

Para os conjuntos de dados Cólon, Glioma e Pâncreas foi utilizado $k = 3$. Para o conjunto de dados de Leucemia, $k = 5$. O número de partições k , para cada conjunto de dados, foi definido conforme o número de exemplos por classe, de maneira a garantir a presença de exemplos de todas as classes em todas as partições. O *kernel* Gaussiano (RBF) foi usado para as SVMs, e portanto, os parâmetros *gamma* (γ) e custo (C) foram ajustados, como mencionado na Seção 5.3.

Neste método experimental, são usadas duas abordagens de referência para comparação com os algoritmos bioinspirados. A primeira, usa os valores padrão adotados pela biblioteca LIBSVM: $\gamma = 1/a$, em que a é o número de atributos do conjunto de dados, e $C = 1$. Essa abordagem será denominada nos experimentos de *Padrão*. A segunda abordagem, denominada neste trabalho de *Grade*, usa uma função da biblioteca LIBSVM, chamada *tune*, para ajustar os parâmetros das SVMs. Essa função usa o *kernel* Gaussiano com validação cruzada via busca de grade (*grid search*) paralela.

Os valores dos parâmetros da SVM são representados diretamente nos algoritmos bioinspirados utilizando codificação real. Nessa codificação, um indivíduo do algoritmo

bioinspirado representa uma solução direta (valores) para os parâmetros. Assim, um indivíduo do algoritmo bioinspirado é formado por dois números reais. O primeiro número representa o parâmetro custo (C) e o segundo representa o parâmetro *gama* (γ). Logo, um indivíduo é formado pelo par ordenado (C, γ) . Neste método experimental, o intervalo definido para os valores do parâmetro $C = [2^{-2}, 2^{12}]$ e do parâmetro $\gamma = [2^{-10}, 2^4]$. Esses intervalos foram estabelecidos com base no trabalho de Hsu et al. (2007).

Para os algoritmos bioinspirados ACO, PSO e AG, são realizadas 100 iterações para para cada execução. Isso implica em 3000 (número de indivíduos multiplicado pelo número de iterações) avaliações da função de aptidão pelo algoritmo PSO e 3030 para os algoritmos ACO e AG. As 30 avaliações a mais devem-se a avaliação da população inicial por esses algoritmos. A avaliação da função de aptidão, para o processo de ajuste de parâmetros, é a execução da validação cruzada. O algoritmo bioinspirado CSA realiza 60 iterações. Esse menor número de iterações é compensado por um maior número de indivíduos (60). O CSA realiza aproximadamente 7000 avaliações da função de aptidão. A razão do alto número de avaliações associadas a esse algoritmo é que ele gera muitos clones, que são avaliados pela função de aptidão.

Resultados do Ajuste de Parâmetros de SVMs com o Método Experimental A

Nas tabelas 5.3, 5.4, 5.5 e 5.6, as taxas de erro de classificação obtidas pelas SVMs com os parâmetros ajustados são apresentadas em porcentagem. O desvio padrão referente às 30 execuções dos algoritmos bioinspirados são apresentados em parênteses, também em porcentagem. Os menores erros estão destacados. As taxas de erro apresentadas foram obtidas pelos classificadores gerados com os parâmetros encontrados pelos algoritmos bioinspirados para as menores taxas de erro de validação.

Tabela 5.3: Cólón - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Cólón (em %).

Algoritmo	Validação	Teste
ACO	15.6 (8.1)	17.3 (7.3)
PSO	17.9 (-)	16.2 (0.9)
CSA	15.9 (2.5)	15.3 (1.5)
AG	16.5 (2.3)	15.6 (1.2)
Grade	13.8 (0.9)	13.0 (0.6)
Padrão	16.2 (-)	11.4 (-)

A abordagem Grade obteve a menor taxa de erro de validação para o conjunto de dados Cólón. Porém, os valores padrão para os parâmetros (abordagem Padrão) obtiveram as menores taxas de erro de teste. Nesse conjunto de dados, ocorreu uma grande diferença entre as taxas de erro de validação e de teste para a abordagem Padrão. A taxa de erro de

Tabela 5.4: Glioma - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Glioma (em %).

Algoritmo	Validação	Teste
ACO	10.2 (-)	16.4 (10.0)
PSO	13.6 (1.3)	15.9 (3.3)
CSA	12.9 (1.8)	15.0 (1.8)
AG	10.5 (1.1)	16.8 (2.2)
Grade	14.6 (2.1)	16.9 (2.8)
Padrão	16.0 (-)	12.0 (-)

Tabela 5.5: Leucemia - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Leucemia (em %).

Algoritmo	Validação	Teste
ACO	10.0 (-)	11.0 (0.3)
PSO	14.5 (11.6)	16.3 (13.8)
CSA	10.0 (4.9)	11.0 (4.0)
AG	33.6 (15.6)	39.1 (18.7)
Grade	12.2 (0.9)	12.3 (0.6)
Padrão	10.9 (-)	11.9 (-)

validação obtida pelas SVMs com os valores padrão foram comparáveis às obtidas pelos algoritmos bioinspirados. Em algumas execuções, o algoritmo ACO obteve taxa de erro de validação e de teste de 12.9%. Contudo, para outras, o algoritmo ficou preso em mínimos locais e não conseguiu taxas de erro inferiores a 35.5%.

Para o conjunto Glioma, todos os algoritmos bioinspirados, especialmente o ACO e o AG, obtiveram taxas de erro de validação menores do que as abordagens de referência. Porém, o mesmo não ocorreu para as taxas de erro de teste, em que a abordagem Padrão conseguiu os melhores resultados. Por causa da diferença considerável entre as taxas de erro de validação e de teste, o desempenho dos algoritmos foi afetado. Esse fato é observado mais facilmente para os algoritmos ACO e AG. Aparentemente, os algoritmos bioinspirados levaram a um *overfitting*.

O ACO e o CSA obtiveram as menores taxas de erro de validação e de teste para o conjunto de dados Leucemia. O AG não conseguiu bons resultados para os conjuntos de validação e de teste. Em algumas execuções, o AG obteve uma taxa de erro de validação de 10.0% e uma taxa de erro de teste de 11.0%. Porém, em outras execuções, ocorreu convergência prematura e o AG ficou preso em mínimos locais e não foi capaz de encontrar taxas de erro menores do que 43.6% para validação e 51.0% para teste. Esse fato levou a uma taxa de erro médio superior a dos outros algoritmos.

Para o conjunto Pâncreas, os algoritmos PSO, CSA e AG apresentaram taxas de erro de validação menores que as abordagens Padrão e Grade. O ACO e a abordagem Grade conseguiram resultados comparáveis, que foram melhores que a abordagem Padrão.

Tabela 5.6: Pâncreas - taxa de erro de validação e de teste e desvio padrão para o conjunto de dados Pâncreas (em %).

Algoritmo	Validação	Teste
ACO	15.0 (8.9)	16.9 (8.3)
PSO	12.3 (-)	14.4 (1.5)
CSA	12.3 (-)	13.9 (1.6)
AG	12.3 (-)	14.3 (1.5)
Grade	14.5 (2.6)	14.9 (1.0)
Padrão	16.5 (-)	16.5 (-)

O CSA obteve os melhores resultados para o conjunto de teste. Entre os algoritmos bioinspirados, apenas o ACO não superou as abordagens Padrão e Grade.

As SVMs obtiveram pequenas taxas de erro para diferentes valores de parâmetros, evidenciando o grande número de mínimos locais presentes no problema de ajuste de parâmetros para esses conjuntos de dados. Na Tabela 5.7, são apresentados os intervalos dos valores para os parâmetros gama (γ) e custo (C) para os melhores resultados obtidos e os valores padrão utilizados para o parâmetro γ . O valor padrão para o parâmetro C é sempre um. É importante observar que, apesar do limite superior do intervalo definido para o parâmetro γ ser 2^4 , o maior valor para os quatro conjuntos de dados é aproximadamente 0.28. O menor valor para esse parâmetro, em dois conjuntos de dados, é o limite inferior do intervalo, 2^{-10} . Por outro lado, os valores encontrados para o parâmetro C normalmente são maiores que 1, com excessão do conjunto Leucemia, e muitas vezes, são próximos do limite superior especificado.

Tabela 5.7: Intervalos de valores encontrados para os parâmetros γ e C .

Conjunto de Dados	γ			C	
	mínimo	máximo	padrão	mínimo	máximo
Cólon	9.77e-04	1.90e-03	3.12e-02	7.94e+00	5.64e+01
Glioma	9.77e-04	2.75e-01	6.25e-02	1.13e+00	3.47e+03
Leucemia	9.32e-03	5.63e-02	1.00e-02	2.50e-01	4.1e+03
Pâncreas	4.34e-02	4.34e-02	1.00e-02	1.72e+02	3.87e+03

Para avaliar se há diferença entre as taxas de erro obtidas, o teste- t pareado corrigido, proposto por Nadeau & Bengio (2003), foi aplicado com 95% de confiança. A hipótese nula é de que não há diferença estatística entre as taxas de erro obtidas.

Para o conjunto Cólon, a hipótese nula foi rejeitada na comparação entre o ACO e a abordagem Padrão. Para o conjunto Glioma e Pâncreas, não houve diferença estatística entre nenhum algoritmo e as abordagens de referência. Para o conjunto Leucemia, houve diferença entre o AG e todos os outros algoritmos e as abordagens de referência.

Adicionalmente, aplicou-se o teste de Friedman (Demšar, 2006), para verificar se há diferença significativa entre os algoritmos/abordagens. O valor crítico da distribuição F

com $\alpha = 0.05$ e 5 $(c - 1)$ e 15 $((c - 1) \times (d - 1))$ graus de liberdade é 2.9, em que c é o número de classificadores e d é o número de conjuntos de dados. A hipótese nula é que não há diferenças significativas entre os algoritmos/abordagens. Como o teste resultou em $F = 1.36$, a hipótese nula não pode ser rejeitada. Em seguida, aplicou-se o mesmo teste para o conjunto de validação, que novamente não pode rejeitar a hipótese nula, com $F = 1.11$. Portanto, todos os algoritmos/abordagens têm desempenhos estatisticamente semelhantes.

5.6 Método Experimental B: Ajuste de Parâmetros de RNs

O ajuste de parâmetros neste método experimental é baseado em dois laços aninhados. O laço interno é usado para determinar a melhor combinação de valores para os parâmetros da RN, ou seja, os valores de parâmetros em que foi obtido o melhor desempenho para o conjunto de dados de validação. O laço externo é usado para estimar o desempenho do classificador gerado com os melhores parâmetros encontrados no laço interno.

A confiança das estimativas obtidas, tanto para o laço interno como para o laço externo, são afetadas pela natureza aleatória dos exemplos usados para treinamento, validação e teste. Para reduzir a variância existente, é usada validação cruzada estratificada para os dois laços. Nestes experimentos, o número de partições para o laço externo (N_D) é igual ao número de partições para o laço interno (N_P), $N_D = N_P = 10$. No laço externo, uma, dentre as N_D partições, é usada como conjunto de teste. No laço interno, uma, dentre as N_P partições, é usada como conjunto de validação. Das nove partições restantes para cada laço, oito são usadas para treinamento e uma é usada para a parada antecipada do treinamento das RNs, ou seja, se o erro obtido para os dados contidos nessa partição começarem a aumentar, o processo de treinamento é interrompido.

Na Figura 5.2 é mostrado um exemplo do uso de validação cruzada para os dois laços aninhados, com $N_D = N_P = 4$. O conjunto de dados D , no laço externo, é dividido em quatro partições: $d1$, $d2$, $d3$ e $d4$. Para formar as partições do laço interno, as partições de treinamento utilizadas no laço externo são juntadas em uma única partição P , que posteriormente, é dividida em 4 partições ($p1$, $p2$, $p3$ e $p4$). A taxa de erro médio obtida no laço interno é a estimativa para a taxa de erro de teste de uma partição do laço externo. No exemplo da Figura 5.2, a taxa de erro médio de validação, 10.3%, é a estimativa para a partição $d1$, em que a taxa de erro é 10%. A taxa de erro de teste para uma solução (combinação de valores para os parâmetros) para o conjunto de dados D é a média da taxa de erro de teste para cada uma das partições do laço externo. Esse valor é 10.9% no exemplo da figura citada. A taxa de erro de validação para o conjunto de dados D , é a média das N_D taxas de erro médio obtidas no laço interno. O objetivo dos algoritmos

bioinspirados é minimizar a taxa de erro de validação para o conjunto de dados D .

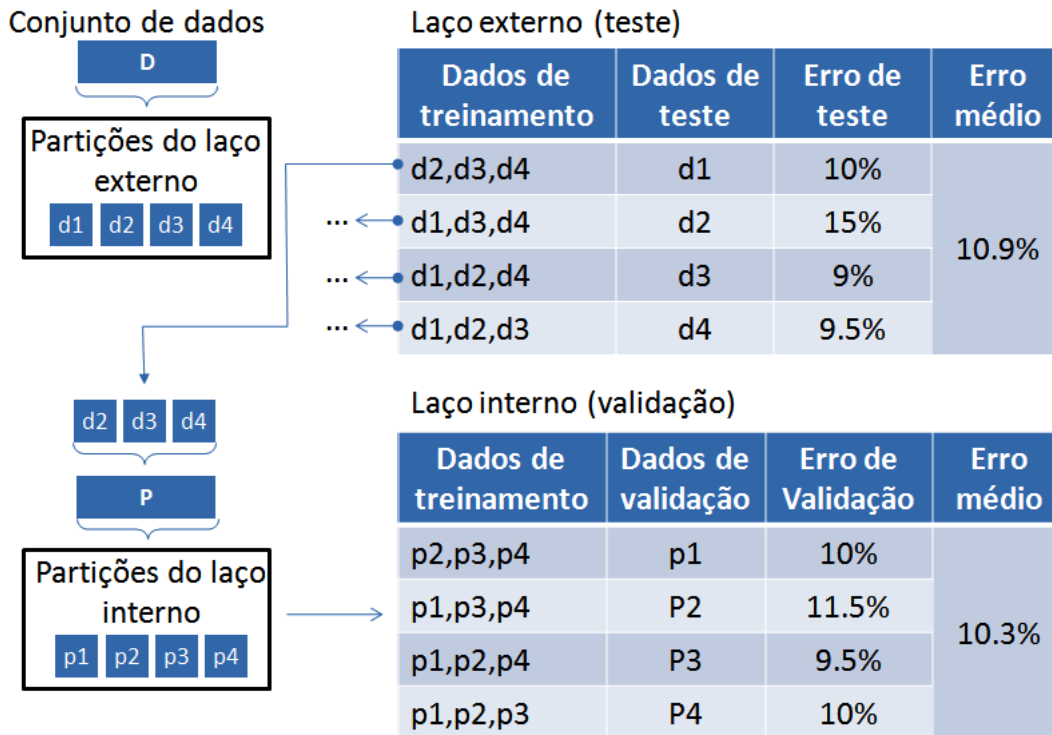


Figura 5.2: Método Experimental B.

A minimização da taxa de erro de validação é realizada pelos algoritmos bioinspirados até que o critério de parada seja satisfeito. Como resposta, os algoritmos bioinspirados fornecem a melhor solução (valores para os parâmetros) obtida, ou seja, a solução com a menor taxa de erro médio obtida no laço interno. Esses valores para os parâmetros são usados para obter a estimativa do erro real, utilizando as partições de teste no laço externo.

Para cada combinação de valores dos parâmetros, o algoritmo de aprendizado é executado $N_D \times N_P$ vezes. Uma combinação de valores para os parâmetros pode resultar em pesos finais diferentes para a rede, usando o mesmo conjunto de dados para treinamento. Isso ocorre porque os pesos da rede foram inicializados aleatoriamente. Para reduzir o efeito dessa variância, adotou-se, neste método experimental, $r = 10$ repetições para cada treinamento realizado. O ideal seriam pelo menos 30 repetições, mas devido ao alto custo computacional foram realizadas apenas 10 repetições de cada treinamento. Assim, o número total de execuções do algoritmo de aprendizado para uma combinação de valores dos parâmetros $E_t = r \times N_D \times N_P$.

O exemplo a seguir tenta esclarecer a idéia do custo computacional envolvido no método experimental proposto. Suponha que um algoritmo de aprendizado possua dois parâmetros a serem ajustados. Suponha ainda, que serão testados 12 diferentes valores para o primeiro parâmetro e 15 diferentes valores para o segundo parâmetro, um total de

180 diferentes possíveis combinações. Se forem usadas 10 partições para os dois laços, ou seja, $N_D = N_P = 10$, e 5 repetições ($r = 5$), o algoritmo de aprendizado será executado $10 \times 5 \times 180 = 9\,000$ vezes para o laço externo e $10 \times 10 \times 5 \times 180 = 90\,000$ vezes para o laço interno. Assim, o algoritmo de aprendizado será executado no total 99\,000 vezes.

Para as RNs, um indivíduo ou solução de um algoritmo bioinspirado representa de maneira direta uma solução para os parâmetros de uma RN. Como foram ajustados três parâmetros para as RNs, conforme descrito na Seção 5.3, um indivíduo é composto por três números reais. O primeiro valor refere-se ao número de neurônios na camada oculta (γ), e pode assumir um valor dentro do intervalo $[2, 100]$. O segundo valor, refere-se ao parâmetro taxa de aprendizado (η), e pode assumir qualquer valor dentro do intervalo $[0.05, 1]$. O terceiro valor refere-se ao parâmetro termo *momentum* (μ) e pode assumir qualquer valor dentro do intervalo $[0, 1]$. Na Figura 5.3 é ilustrado um exemplo de um indivíduo com $\gamma = 36$, $\eta = 0.1$ e $\mu = 0.8$.

γ	η	μ
36	0.1	0.8

Figura 5.3: Representação de um indivíduo para ajuste de parâmetros de RNs.

Os resultados obtidos são armazenados em uma estrutura de dados apropriada, explicada a seguir. Com essa estrutura, todos os algoritmos bioinspirados usados nos experimentos podem acessar os resultados rapidamente, evitando executar o algoritmo de aprendizado milhares de vezes para cada algoritmo bioinspirado. Para que o armazenamento seja possível, os intervalos definidos para os parâmetros foram discretizados, e os valores encontrados pelos algoritmos bioinspirados são arredondados para o valor mais próximo. Os valores para os parâmetros η e μ foram arredondados para frações de 0.05 e 0.1, respectivamente. Assim, o domínio dos valores para o parâmetro η é $\{0.05, 0.1, 0.15, \dots, 1\}$, e para o parâmetro μ o domínio é $\{0, 0.1, 0.2, \dots, 1\}$. Adicionalmente, os valores para o parâmetro γ foram arredondados para valores pares, $\{2, 4, \dots, 100\}$. Assim, são possíveis 11\,000 diferentes combinações de valores para os parâmetros.

Neste método experimental, foram propostas duas abordagens de referência para o ajuste de parâmetros de RNs. A primeira, usa os valores padrão adotados pelo WEKA para RNs MLP, ou seja, $\gamma = (A + F)/2$, $\eta = 0.3$ e $\mu = 0.2$, em que A e F são o número de atributos e o número de classes, respectivamente. Essa abordagem é referida como *Padrão*. A segunda abordagem de referência consiste em gerar aleatoriamente s diferentes combinações de valores para os parâmetros, em que s é o número aproximado de indivíduos testados pelos algoritmos bioinspirados. A melhor combinação, ou seja, com a menor taxa de erro de validação, é usada para a geração dos classificadores no laço externo. Essa abordagem é referida neste trabalho como *Aleatória*.

Como mencionado anteriormente, o número de iterações foi definido como o critério de parada para os algoritmos bioinspirados. Neste método experimental, os algoritmos

bioinspirados ACO, AG e PSO param após 100 iterações. O algoritmo CSA, pára após 60 iterações. Após o término das iterações os algoritmos retornam os parâmetros que resultaram no classificador com a menor taxa de erro de validação. O número de iterações estabelecidas para o CSA é menor que os demais algoritmos, pois o CSA realiza aproximadamente o dobro de avaliações da função de aptidão devido ao processo de clonagem, em que cada clone precisa ser avaliado pela função de aptidão. Nestes experimentos, assim como na seção anterior, foram usados quatro conjuntos de dados de expressão gênica a saber: Cólon, Glioma, Leucemia e Pâncreas.

Armazenamento dos Dados

Para armazenar os resultados obtidos, foi utilizado um *array* multi-dimensional. Os resultados do laço externo foram armazenados em um *array* de seis dimensões. A seguir, é descrito o que é armazenado em cada dimensão:

1^a : número da partição de teste;

2^a : parâmetro referente ao número de neurônios na camada oculta (γ);

3^a : parâmetro referente à taxa de aprendizado (η);

4^a : parâmetro referente ao termo *momentum* (μ);

5^a : número da repetição (r);

6^a : medidas de desempenho (erro, precisão, *recall* e Medida-F).

Suponha que se deseja acessar o erro de classificação armazenado em um *array* denominado *resultados* com os seguintes valores para as dimensões de um a cinco: 3, 36, 0.1, 0.8 e 4, respectivamente. Esse erro pode ser acessado da seguinte maneira:

$erro = resultado(3, 36, 0.1, 0.8, 4, "erro")$.

Os resultados do laço interno para RNs foram armazenados utilizando um *array* com uma dimensão a mais em relação ao laço externo. Essa dimensão especifica o número da partição de validação do laço interno. Assim, é possível acessar o erro obtido para qualquer partição de validação.

Resultados do Ajuste de Parâmetros de RNs com o Método Experimental B

Primeiramente, antes de utilizar os algoritmos bioinspirados para o ajuste de parâmetros, foi verificado se há correlação entre as taxas de erro de validação e de teste. Isso é possível pois os resultados de classificação para todas as combinações de parâmetros estão armazenados. A correlação entre duas variáveis reflete o grau que essas variáveis estão

relacionadas. O coeficiente de correlação de Pearson (r), ou simplesmente correlação de Pearson, reflete o grau de relação linear entre duas variáveis. O coeficiente r pode assumir valores entre -1 e 1 . Quando $r = 1$, significa que há uma correlação positiva perfeita entre as duas variáveis. Quando $r = -1$, significa que há uma correlação negativa perfeita entre as duas variáveis, ou seja, grandes valores de uma variável estão associados a pequenos valores da outra variável. Quando $r = 0$, significa que as duas variáveis são absolutamente independentes. Porém, como a correlação de Pearson avalia apenas a correlação entre variáveis lineares, pode existir uma dependência não linear e outros meios devem ser investigados. A correlação de Pearson foi aplicada para os quatro conjuntos de dados. Para os conjuntos Cólón, Glioma, Leucemia e Pâncreas a correlação (r) foi 0.87, 0.97, 0.74 e 0.89, respectivamente. Para os conjuntos de dados Cólón, Glioma e Pâncreas, os algoritmos bioinspirados poderão guiar suas buscas confiantes nas estimativas de validação obtidas. A correlação para o conjunto de dados Leucemia foi um pouco mais baixa. Porém, essa correlação ainda é suficiente para guiar as buscas dos algoritmos.

Nas Tabelas 5.8, 5.9, 5.10 e 5.11 são apresentados os resultados de classificação obtidos pelas RNs com os parâmetros ajustados por cada algoritmo bioinspirado. Nessas tabelas são apresentadas, em porcentagem, as taxas de erro médio e o desvio padrão (em parênteses) referente às 30 execuções dos algoritmos bioinspirados para os conjuntos de validação e de teste. As menores taxas de erro estão destacadas. Adicionalmente, o número médio de diferentes combinações (soluções) são apresentados. Esse número corresponde à média das 30 execuções realizadas por cada algoritmo/abordagem. A abordagem Padrão sempre utiliza a mesma combinação de valores para um conjunto de dados. Porém, ao contrário das SVMs, pode ocorrer diferenças devido à inicialização aleatória dos pesos da rede.

Tabela 5.8: Cólón - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Cólón.

Algoritmo	Validação	Teste	‡ Comb.
ACO	15.2 (0.0)	15.6 (0.8)	1353
AG	15.4 (0.2)	13.8 (1.2)	221
CSA	15.6 (0.3)	13.4 (1.4)	1361
PSO	15.3 (0.1)	12.9 (0.9)	1200
Aleatória	15.4 (0.1)	15.2 (1.8)	2595
Padrão	19.7 (0.0)	17.9 (0.0)	◇

A primeira observação que pode ser feita é que os algoritmos bioinspirados obtiveram resultados melhores ou semelhantes às abordagens Aleatória e Padrão. A menor taxa de erro de validação e de teste, para todos os conjuntos de dados, foi obtido por um algoritmo bioinspirado. Apenas para o conjunto de dados Glioma, para a taxa de erro de validação, é que a abordagem Aleatória conseguiu resultados iguais aos dos algoritmos ACO, AG e PSO, e um pouco menor do que o CSA. Isso indica que os algoritmos estão encontrando boas combinações de valores para os parâmetros.

Tabela 5.9: Glioma - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Glioma.

Algoritmo	Validação	Teste	‡ Comb.
ACO	14.9 (0.0)	15.0 (0.0)	1461
AG	14.9 (0.1)	15.0 (0.1)	228
CSA	15.0 (0.1)	14.8 (0.4)	1298
PSO	14.9 (0.0)	15.0 (0.0)	1199
Aleatória	14.9 (0.0)	15.0 (0.2)	2600
Padrão	17.5 (0.0)	15.7 (0.0)	◇

Tabela 5.10: Leucemia - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Leucemia.

Algoritmo	Validação	Teste	‡ Comb.
ACO	13.9 (0.2)	15.0 (0.2)	1533
AG	15.3 (0.5)	16.2 (0.8)	249
CSA	15.3 (0.4)	15.8 (0.8)	1362
PSO	14.7 (0.6)	15.7 (0.4)	1633
Aleatória	14.6 (0.4)	15.5 (0.5)	2599
Padrão	18.4 (0.0)	22.1 (0.0)	◇

A comparação entre os diferentes algoritmos bioinspirados mostra que, em geral, o ACO encontrou os melhores resultados. Esse algoritmo conseguiu as menores taxas de erro de validação para os quatro conjuntos de dados. As taxas de erro de teste foram menores para os conjuntos de Leucemia e Pâncreas, e similares para o conjunto Glioma. Porém, para o conjunto Cólon, o ACO foi o pior algoritmo bioinspirado. O desempenho do ACO, relativamente às outras técnicas, não é tão bom nos conjuntos de dados em que a diferença entre o algoritmo bioinspirado e a abordagem Padrão é pequena. Isso pode indicar que há uma grande quantidade de combinações de parâmetros que resultam em taxas de erro mínimas para esses conjuntos de dados.

Alguns testes foram executados para avaliar a significância estatística das diferenças observadas. O primeiro teste aplicado foi o teste-*t* pareado corrigido, com nível de signi-

Tabela 5.11: Pâncreas - taxa de erro de validação e de teste, desvio padrão (em %) e número de diferentes combinações testadas para o conjunto de dados Pâncreas.

Algoritmo	Validação	Teste	‡ Comb.
ACO	15.4 (0.4)	15.7 (0.9)	1486
AG	16.6 (0.4)	17.6 (0.8)	223
CSA	16.5 (0.4)	17.6 (1.5)	1391
PSO	16.1 (0.5)	17.4 (1.4)	2151
Aleatória	16.1 (0.3)	17.5 (1.6)	2601
Padrão	17.8 (0.0)	17.3 (0.0)	◇

ficância $\alpha = 0.05$ (95% de confiança) e nove graus de liberdade, correspondendo às $k - 1$ partições. Os dados usados foram as taxas de erro para cada uma das 10 partições. A hipótese nula é que os algoritmos e as abordagens conseguiram o mesmo desempenho. De acordo com a distribuição de *Student*, quando a comparação entre dois algoritmos resultar em um *p-value* maior do que 1.833, pode-se rejeitar a hipótese nula. O teste estatístico mostrou que não há diferenças significativas para as taxas de erro de teste. Porém, para as taxas de erro de validação, foram observadas algumas diferenças significativas. Como podem ser observadas nas tabelas apresentadas, as taxas de erro de validação, geralmente, variam mais do que as de teste, o que justifica a rejeição da hipótese nula. Os resultados para esse teste são apresentados na Tabela 5.12. Cada célula $C_{i,j}$ representa a comparação entre o algoritmo/abordagem da linha i com o algoritmo/abordagem da coluna j . Cada célula possui a inicial referente ao conjunto de dados em que o algoritmo/abordagem i é significativamente diferente do algoritmo/abordagem j : **{C}**ólon, **{G}**lioma, **{L}**euçemia e **{P}**âncreas. Um hífen é usado quando a diferença não é significativa.

Esses resultados normalmente confirmam as observações anteriores. Há, contudo, algumas diferenças. Para os conjuntos de dados Cólón, Glioma e Leucemia, as diferenças entre os algoritmos bioinspirados e a abordagem Padrão é significativa. Porém, para o conjunto de dados Pâncreas, o ACO foi o único algoritmo com diferença significativa para a abordagem Padrão. Para o conjunto de dados Leucemia, o ACO foi significativamente diferente de todos os outros algoritmos e abordagens, com exceção do PSO.

Tabela 5.12: Análise estatística - teste-*t* pareado corrigido (conjunto de dados de validação).

	AG	CSA	PSO	Aleatória	Padrão
ACO	- - L -	- - L -	- - - -	- - L -	C G L -
AG		- - - -	- - L -	- - L -	C G L P
CSA			- - L -	- - L -	C G L -
PSO				- - - -	C G L -
Aleatória					C G L -

Adicionalmente aplicou-se o teste de Friedman (Demšar, 2006), para verificar se há diferença significativa entre qualquer um dos algoritmos ou abordagens, considerando as taxas de erro de teste de todos os conjuntos de dados. O valor crítico da distribuição F com $\alpha = 0.05$ e 5 $(c - 1)$ e 15 $((c - 1) \times (d - 1))$ graus de liberdade é 2.9, em que c é o número de classificadores e d é o número de conjuntos de dados. Como o teste resultou em $F = 1.04$, a hipótese nula, de que não há diferença estatística entre os resultados obtidos pelos algoritmos e as abordagens, foi aceita. Assim, aplicou-se o mesmo teste para o conjunto de validação, que resultou na rejeição da hipótese nula, com $F = 11.74$. Em seguida, aplicou-se o teste de Nemenyi (Demšar, 2006) para identificar quais pares eram significativos. Na Tabela 5.13 é apresentado o *ranking* médio usando o teste de

Friedman para as taxas de erro de validação e de teste. O valor crítico para comparar o *ranking*-médio de dois algoritmos/abordagens com $\alpha = 0.05$ é 3.77. O resultado do teste confirmou que o ACO foi significativamente melhor do que a abordagem Padrão no conjunto de validação.

Tabela 5.13: *Ranking* Médio usando o teste de Friedman.

Algoritmo	Validação	Teste
ACO	1.375	2.625
AG	3.875	4.250
CSA	4.625	3.125
PSO	2.500	2.625
Aleatória	2.625	3.375
Padrão	6.000	5.000

Os números de diferentes combinações testadas por cada algoritmo, apresentados nas tabelas de 5.8 a 5.11, provêm informação sobre o compromisso entre exploração e prospecção. Como mencionado anteriormente, o número máximo de indivíduos que cada algoritmo é capaz de testar é determinado pelos seus parâmetros. Nestes experimentos, os algoritmos ACO, PSO e AGs poderiam testar aproximadamente 3 000 indivíduos, enquanto que o algoritmo CSA, devido ao processo de clonagem, poderia testar aproximadamente 7 000 indivíduos. Com os intervalos definidos, são possíveis 11 000 combinações para os valores dos parâmetros. A proporção de diferentes combinações testadas pelos algoritmos ACO, PSO e CSA indica um bom balanço entre exploração e prospecção do espaço de busca. Exploração é a habilidade de testar várias regiões do espaço de busca para localizar uma região promissora, preferencialmente a ótima. Prospecção é a habilidade de concentrar a busca em uma região promissora de forma a melhorar o resultado obtido até o momento para essa região (Trelea, 2003). Por outro lado, o AG explorou uma proporção menor de diferentes combinações. Isso ocorreu mesmo sem o uso do elitismo, processo que pode aumentar a prospecção. Isso pode ter ocorrido devido aos valores dos parâmetros usados para o algoritmo. Provavelmente, um aumento na probabilidade de mutação minimizasse esse problema. Apesar disso, os resultados obtidos pelo AG foram comparáveis aos resultados obtidos por outros algoritmos. Isso significa que não há uma correlação clara entre o total de exploração e a qualidade das soluções obtidas. Como mencionado na Seção 4.2, o elitismo não foi usado em razão dos resultados obtidos durante alguns experimentos realizados com funções de *benchmark*.

Na Tabela 5.14, são apresentadas as médias de algumas medidas referente aos valores encontrados para os parâmetros, considerando todas as execuções dos algoritmos bioinspirados. O número de neurônios utilizados pela abordagem Padrão para os conjuntos de dados Cólon, Glioma, Leucemia e Pâncreas é 9, 17, 26 e 26, respectivamente. O parâmetro η é igual a 0.3 e o parâmetro μ é igual a 0.2 para todos os conjuntos de dados. Como

é possível observar, os valores encontrados para os parâmetros abrangem quase todo o espaço de busca delimitado. Para o conjunto de dados Leucemia e Pâncreas, o número de neurônios ocultos encontrados é pequeno se comparado aos outros dois conjuntos. O menor valor desse parâmetro, para o conjunto Glioma, foi 82. Para o conjunto Leucemia, o valor mínimo para o parâmetro termo *momentum* ficou acima de 0.7. As medianas das taxas de aprendizado ficaram acima de 0.6 para os conjuntos Cólon e Glioma, e abaixo de 0.2 para o conjunto Pâncreas.

Tabela 5.14: Estatísticas dos melhores valores de parâmetros encontrados para o conjunto de validação para todos os algoritmos bioinspirados.

Parâmetro	Medida	Cólon	Glioma	Leucemia	Pâncreas
γ (nr.neurônios)	Máximo	98	99	53	80
	Mediana	65	92	18	17
	Mínimo	2	82	5	3
η	Máximo	0.95	0.93	0.86	0.83
	Mediana	0.65	0.69	0.46	0.19
	Mínimo	0.21	0.23	0.05	0.05
μ	Máximo	0.93	0.83	1.00	1.00
	Mediana	0.48	0.60	0.95	0.75
	Mínimo	0.18	0.50	0.78	0.08

Como os resultados para todos os valores dos parâmetros foram armazenados, é possível extrair algumas informações. Na Tabela 5.15 são apresentadas a mediana, o mínimo e o máximo para cada conjunto de dados. Esses valores são as médias para todas as partições de todas as combinações de valores dos parâmetros. É possível observar que os algoritmos bioinspirados e a abordagem Aleatória conseguiram taxas de erro de validação mínimas ou próximas das mínimas, dentre todas as combinações testadas. As taxas de erro medianas mostram que há uma grande quantidade de combinações de valores para os parâmetros que resultam em pequenas taxas de erro, apesar de existirem combinações com taxas de erro muito elevadas.

Tabela 5.15: Medidas da taxa de erro de cada conjunto de dados (média para as 10 partições).

Medida	Val./Teste	Cólon	Glioma	Leucemia	Pâncreas
Máximo	Validação	35.2	55.5	49.4	45.8
	Teste	35.2	55.5	49.4	45.8
Mediana	Validação	19.2	16.8	18.7	17.7
	Teste	15.9	15.0	22.3	17.3
Mínimo	Validação	15.2	14.9	13.9	15.2
	Teste	10.5	11.1	11.9	12.8

5.7 Método Experimental C: Ajuste de Parâmetros de RNs e SVMs

Este método experimental, assim como o método B, usa dois laços aninhados. A única diferença é que neste método, os parâmetros são ajustados para cada partição do laço externo, ou seja, a minimização do erro de validação é feita independentemente para cada partição do laço externo. A modificação do método experimental ocorreu para que os dados usados para teste no laço externo não fossem usados indiretamente no processo de ajuste de parâmetros. Isso garante que os algoritmos bioinspirados utilizam estimativas calculadas apenas com os dados de treinamento para cada partição do laço externo. O método adotado nestes experimentos foi utilizado também em Statnikov et al. (2005b) e Statnikov et al. (2005a).

Suponha que exista apenas um parâmetro α a ser ajustado para um algoritmo de aprendizado A, e que α possa assumir m diferentes valores: $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_m$. O desempenho D_i de um classificador treinado pelo algoritmo de aprendizado A com o parâmetro α_i é estimado para $i = 1, \dots, m$ no laço interno. O modelo final é construído treinando o algoritmo A com o parâmetro α_{melhor} no laço externo, em que $melhor = \operatorname{argmax}(D_i)$, para $i = 1, 2, \dots, m$. Assim como no método experimental B, N_D e N_P são o número de partições para o laço externo e interno, respectivamente. O Algoritmo 5 é usado para implementar este método experimental.

Este método para ajuste de parâmetros é muito custoso computacionalmente. Nestes experimentos é investigado também o efeito de reduzir o número de execuções para o laço interno. Isso é feito para as RNs, que possuem um custo computacional 10 vezes maior do que as SVMs, por realizarem 10 repetições para cada treinamento ($r = 10$). O que é avaliado é o compromisso entre a perda da qualidade das taxas de erro de validação e o ganho obtido buscando mais soluções no tempo economizado. Portanto, é avaliado o efeito da substituição da média dos resultados de $r \times N_P$ com os resultados de uma simples rodada ($r = 1$) dos algoritmos para apenas uma partição ($N_P = 1$). O processo que utiliza a média das 10 repetições ($r = 10$) e das 10 partições ($N_P = 10$) será denominado 10/10, enquanto que o processo que utiliza apenas uma repetição para uma partição, será denominado de 1/1.

A codificação adotada para as RNs é a mesma do método experimental B. Como dito anteriormente, há 11 000 diferentes combinações de valores para os parâmetros. O número de iterações dos algoritmos bioinspirados para o processo 10/10 é o mesmo do método experimental A e B. Para o processo 1/1, o número de indivíduos foi dobrado e, conseqüentemente, é possível analisar o dobro de combinações de valores para os parâmetros.

Neste método, a abordagem de codificação adotada para representar os valores de parâmetros de SVMs utiliza potências para a base 2 (Hsu et al., 2007), ou seja, para encontrar os valores dos parâmetros C e γ os valores representados pelo indivíduo são

Algoritmo 5 Método experimental C para ajuste de parâmetros.

```

 $D \leftarrow$  conjunto de dados
 $l \leftarrow 1$ 
enquanto  $l \leq N_D$  faça
   $conjunto\_treino \leftarrow (N_D - 1)$  partições de  $D$ 
   $conjunto\_teste \leftarrow$  partição restante de  $D$ 
   $i \leftarrow 1$ 
  enquanto  $i \leq m$  faça
     $n \leftarrow 1$ 
     $P \leftarrow conjunto\_treino$ 
    enquanto  $n \leq N_P$  faça
       $conjunto\_treino\_validação \leftarrow (N_P - 1)$  partições de  $P$ 
       $conjunto\_teste\_validação \leftarrow$  partição restante de  $P$ 
       $C \leftarrow$  Treinar A com  $conjunto\_treino\_validação$  com parâmetro  $\alpha_i$ 
       $P(n) \leftarrow$  Testar classificador  $C$  para o  $conjunto\_teste\_validação$ 
       $n \leftarrow n + 1$ 
    fim enquanto
     $D(i) \leftarrow média(P)$ 
     $i \leftarrow i + 1$ 
  fim enquanto
   $\alpha_{melhor} \leftarrow argmax(D(i))$ 
   $M \leftarrow$  Treinar A com  $conjunto\_treino$  usando  $\alpha_{melhor}$ 
   $\rho(l) \leftarrow$  Testar classificador  $M$  para  $conjunto\_teste$ 
   $l \leftarrow l + 1$ 
fim enquanto
retornar  $média(\rho)$ 

```

potências para a base 2. Caso o primeiro valor do indivíduo seja 1 e o segundo seja -3 , por exemplo, $C = 2^1$ e $\gamma = 2^{-3}$. O intervalo definido para o parâmetro $C = [-5, 15]$, e para o parâmetro $\gamma = [-15, 3]$, ambos discretizados em frações de 1. Assim, há 399 diferentes combinações de valores que podem ser usadas para os parâmetros. Como há um pequeno número de soluções, os algoritmos bioinspirados realizaram apenas 20 iterações com cinco indivíduos na população. Essas configurações possibilitaram que os algoritmos explorassem no máximo 100 soluções para os parâmetros. Conseqüentemente, a abordagem Aleatória podia testar no máximo 100 combinações de valores.

Armazenamento dos Dados

Neste método experimental, as RNs foram aplicadas a quatro conjuntos de dados. Os resultados armazenados gerados nos experimentos B, para RNs, foram utilizados novamente nestes experimentos. Porém, esses dados foram usados de maneira diferente do uso feito no método experimental B, como descrito no início desta seção. Os 10 conjuntos de dados descritos na Tabela 5.1 foram utilizados nos experimentos com SVMs. Os resultados obtidos foram armazenados em um *array* multi-dimensional, similar ao usado para as

RNs. Os resultados do laço externo foram armazenados utilizando um *array* de quatro dimensões. A seguir, é descrito o que é armazenado em cada dimensão:

1^a : número da partição de teste;

2^a : parâmetro custo (C);

3^a : parâmetro *gama* (γ);

4^a : medidas de desempenho (erro, precisão, *recall* e *Medida-F*) e número de vetores de suporte.

Suponha que se deseja acessar o erro de classificação armazenado em uma estrutura de dados como a apresentada, sendo esse *array* denominado *resultados*. Se o número da partição de teste é 5, $C = 1$ e $\gamma = 0.01$, então esse erro pode ser acessado da seguinte maneira:

$erro = resultado(5, 1, 0.01, "erro")$.

Os resultados do laço interno para SVMs foram armazenados utilizando um *array* de cinco dimensões, que possui uma dimensão a mais em relação ao laço externo, para especificar o número da partição de validação do laço interno. Assim, é possível acessar a taxa de erro obtida para qualquer partição.

5.7.1 Resultados do Ajuste de Parâmetros de RNs com o Método Experimental C

Nas Tabelas de 5.16 a 5.19 são apresentadas as taxas de erro (em porcentagem) de classificação para os conjuntos de validação e de teste usando as RNs geradas com os valores dos parâmetros encontrados por cada algoritmo bioinspirado. No processo de estimação denominado 10/10 foram usadas 10 partições ($k = 10$) e 10 repetições ($r = 10$). Os valores entre parênteses são os valores de desvio padrão para as 30 execuções dos algoritmos bioinspirados. Além disso é apresentado o número de diferentes soluções testadas para os parâmetros das RNs MLP. As menores taxas de erro estão destacadas. Nas Tabelas de 5.20 a 5.23 são apresentados os resultados para o processo de estimação denominado 1/1 ($r = 1$ e $N_P = 1$) Para esse processo o número de partições para o laço externo se manteve, ou seja, $N_D = 10$.

Análise dos Resultados com o Processo de Estimação 10/10

É possível observar que os algoritmos bioinspirados geralmente obtiveram melhores resultados do que a abordagem Padrão. Isso significa que a busca feita por esses algoritmos está, de fato, trabalhando como esperado. Por outro lado, os resultados são similares aos obtidos pela abordagem Aleatória. Isso indica que encontrar bons valores para os parâmetros das RNs para esses conjuntos de dados não é difícil. Esse fato pode ser

Tabela 5.16: Cólón - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Cólón.

Algoritmo	Validação	Teste	# Comb.
ACO	12.7 (0.1)	18.1 (1.0)	1546
AG	13.7 (0.2)	16.6 (1.6)	253
CSA	13.9 (0.2)	15.8 (1.8)	1425
PSO	13.5 (0.1)	16.5 (1.7)	1517
Aleatória	13.2 (0.2)	16.7 (1.6)	2623
Padrão	19.8 (-)	17.9 (-)	◇

Tabela 5.17: Glioma - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Glioma.

Algoritmo	Validação	Teste	# Comb.
ACO	12.3 (0.2)	16.0 (1.7)	1552
AG	13.0 (0.1)	14.5 (0.9)	317
CSA	12.8 (0.3)	17.5 (2.5)	1481
PSO	12.8 (0.1)	14.0 (1.3)	1477
Aleatória	12.6 (0.2)	17.5 (2.1)	2622
Padrão	18.0 (-)	15.8 (-)	◇

Tabela 5.18: Leucemia - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Leucemia.

Algoritmo	Validação	Teste	# Comb.
ACO	12.8 (0.1)	20.3 (0.9)	1608
AG	13.9 (0.3)	20.3 (0.9)	256
CSA	13.9 (0.2)	19.5 (1.3)	1384
PSO	13.3 (0.2)	19.7 (1.0)	1557
Aleatória	13.2 (0.1)	19.2 (1.2)	2623
Padrão	18.9 (-)	23.0 (-)	◇

Tabela 5.19: Pâncreas - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Pâncreas.

Algoritmo	Validação	Teste	# Comb.
ACO	12.4 (0.2)	21.6 (1.5)	1615
AG	14.3 (0.5)	18.9 (1.8)	416
CSA	14.0 (0.2)	20.3 (1.8)	1536
PSO	13.4 (0.3)	19.0 (1.6)	1775
Aleatória	13.3 (0.2)	20.0 (1.7)	2623
Padrão	17.8 (-)	17.3 (-)	◇

Tabela 5.20: Cólón - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Cólón.

Algoritmo	Validação	Teste	# Comb.
ACO	0.1 (0.3)	19.7 (2.2)	8798
AG	1.1 (1.2)	17.3 (2.7)	4426
CSA	0.0 (0.0)	18.3 (3.1)	893
PSO	1.1 (1.0)	17.5 (2.4)	1289
Aleatória	0.0 (0.0)	17.5 (1.7)	9987
Padrão	18.3 (-)	17.9 (-)	◇

Tabela 5.21: Glioma - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Glioma.

Algoritmo	Validação	Teste	# Comb.
ACO	2.0 (0.0)	22.1 (3.3)	8800
AG	3.2 (1.0)	19.3 (2.6)	4488
CSA	2.0 (0.0)	19.3 (3.4)	1452
PSO	2.9 (1.0)	19.7 (3.4)	1532
Aleatória	2.0 (0.0)	19.4 (3.6)	9986
Padrão	16.0 (-)	15.8 (-)	◇

Tabela 5.22: Leucemia - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Leucemia.

Algoritmo	Validação	Teste	# Comb.
ACO	1.0 (0.0)	21.2 (1.8)	8792
AG	1.0 (0.0)	22.8 (2.9)	4584
CSA	1.0 (0.0)	22.1 (2.3)	739
PSO	1.0 (0.0)	22.8 (1.9)	894
Aleatória	1.0 (0.0)	22.6 (2.1)	9988
Padrão	9.0 (-)	23.0 (-)	◇

Tabela 5.23: Pâncreas - taxa de erro de validação e de teste (em %), desvio padrão e o número de diferentes combinações testadas para o conjunto de dados Pâncreas.

Algoritmo	Validação	Teste	# Comb.
ACO	0.7 (0.8)	19.5 (2.1)	8796
AG	1.4 (0.6)	20.5 (2.0)	5408
CSA	0.5 (0.8)	20.9 (2.3)	1159
PSO	1.2 (0.7)	21.3 (2.2)	1471
Aleatória	0.4 (0.7)	21.6 (2.4)	9989
Padrão	19.7 (-)	17.3 (-)	◇

confirmado pelos gráficos de A.1 a A.4, exibidos no Apêndice A. Esses gráficos mostram que há uma grande densidade no início das curvas, ou seja, muitas combinações de valores para os parâmetros resultam em pequenas taxas de erro de validação e de teste. A comparação entre os diferentes algoritmos bioinspirados e as abordagens de referência mostra que o ACO encontrou os melhores resultados. Esse algoritmo obteve as menores taxas de erro de validação para todos os conjuntos de dados. Porém, não é possível identificar o melhor algoritmo em termos das taxas de erro de teste.

Como pode ser observado, os parâmetros que resultaram nas menores taxas de erro de validação não resultaram nas menores taxas de erro de teste. Essa diferença pode ser explicada pela baixa correlação que existe entre as taxas de erro de validação e de teste no método adotado para a realização destes experimentos. Essa correlação era maior para o método adotado nos experimentos B. Porém, essa melhor correlação ocorreu porque estimativas do erro de teste eram embutidas indiretamente no processo de ajuste dos parâmetros, como explicado na Seção 5.6, o que não acontece com estes experimentos. Na Tabela 5.24, são mostradas as correlações entre as taxas de erro de validação e de teste para cada conjunto de dado.

Tabela 5.24: Correlação das taxas de erro de validação e de teste para todas as combinações de parâmetros.

Conjunto de dados	Média	Desvio Padrão
Cólon	0.436	0.322
Glioma	0.715	0.159
Leucemia	0.498	0.360
Pâncreas	0.626	0.288

Como dito anteriormente, o número máximo de indivíduos que cada algoritmo é capaz de testar é determinado pelos seus parâmetros. Nestes experimentos, o ACO, PSO e AG poderiam testar aproximadamente 3 000 soluções, enquanto o CSA, devido ao processo de clonagem proporcional à aptidão dos indivíduos, poderia testar aproximadamente 7 000 soluções. Há 11 000 diferentes combinações possíveis de valores para os parâmetros. Assim como no método experimental B, a proporção de diferentes combinações testadas pelo ACO, PSO e CSA indica um bom balanço entre exploração e prospecção do espaço de busca. Por outro lado, o AG explorou uma proporção muito menor de combinações. Como mencionado anteriormente, o elitismo, o qual aumenta a prospecção, não foi usado. Apesar disso, os resultados do AG são comparáveis aos obtidos por outros algoritmos, e assim como nos método experimental B, não há uma relação clara entre o total de exploração e a qualidade das soluções obtidas. Novamente isso pode ser atribuído ao fato de não ser difícil encontrar bons valores para os parâmetros das RNs para esses conjuntos de dados.

Efeito da Simplificação do Processo de Estimação

Nas Tabelas de 5.20 a 5.23 é possível observar que as taxas de erro de validação com o processo 1/1 são, em geral, muito menores do que as obtidas com o processo 10/10. De fato, essas taxas de erro são quase zero, o que indica que está ocorrendo *overfitting*. Como foi realizada a otimização usando apenas uma partição e uma repetição, as pequenas taxas de erro de validação encontradas não correspondem às taxas de erro de teste, que foram muito maiores, inclusive maiores do que os erros de teste encontrados usando o processo 10/10.

De acordo com esses resultados, há um compromisso entre a qualidade das estimativas de validação e o número de soluções testadas. Porém, como um grande espaço de busca já é explorado, é muito mais importante obter estimativas mais precisas. Assim, as próximas análises e testes estatísticos reportados são referentes apenas ao processo 10/10.

Testes Estatísticos

Assim como nos experimentos anteriores, foram realizados testes estatísticos para avaliar as diferenças observadas. Para isso, novamente foram usadas as taxas de erro obtidas pelos classificadores. Inicialmente, foi aplicado o teste-*t* pareado corrigido, com nível de significância $\alpha = 0.05$ (95% de confiança) e nove graus de liberdade, correspondendo às $k - 1$ partições. A hipótese nula é que os algoritmos e as abordagens conseguiram o mesmo desempenho. O teste estatístico mostrou que não há diferenças significativas para as taxas de erro de teste. Porém, quando aplicado para as taxas de erro de validação, algumas diferenças significativas foram observadas. Os resultados para o teste estatístico realizado são apresentados na Tabela 5.25. Cada célula $C_{i,j}$ representa a comparação entre o algoritmo/abordagem da coluna i com o algoritmo/abordagem da coluna j . Cada célula possui a inicial referente ao conjunto de dados em que o algoritmo/abordagem i é significativamente diferente do algoritmo/abordagem j : **{C}**ólon, **{G}**lioma, **{L}**eucremia e **{P}**âncreas. Um hífen é usado quando a diferença não é significativa.

Tabela 5.25: Análise estatística - teste-*t* pareado corrigido (conjunto de dados de validação).

	AG	CSA	PSO	Aleatória	Padrão
ACO	C - L P	C - L P	C - L P	C - - P	C G L P
AG		- - - P	- G L P	C G L P	C G L P
CSA			C - L P	C - L P	C G L P
PSO				- - - -	C G L P
Aleatória					C G L P

Com os resultados dos testes estatísticos, pode-se afirmar que realizar o ajuste de parâmetros pode reduzir as taxas de erro de validação. Porém, essa melhoria não foi grande o suficiente para refletir nas taxas de erro de teste.

Para estes experimentos, também aplicou-se o teste de Friedman (Demšar, 2006) para verificar se há diferença significativa entre os algoritmos/abordagens para todos os conjun-

tos de dados. O valor crítico da distribuição F com $\alpha = 0.05$ e 5 $(c-1)$ e 15 $((c-1) \times (d-1))$ graus de liberdade é 2.9, em que c é o número de classificadores e d é o número de conjuntos de dados. Como o teste resultou em $F = 1.42$, a hipótese nula, de que não há diferença estatística entre os resultados obtidos pelos algoritmos e as abordagens, foi aceita. Aplicou-se o mesmo teste para o conjunto de validação, que resultou na rejeição da hipótese nula, com $F = 64.2$. Em seguida, aplicou-se o teste de Nemenyi (Demšar, 2006) para identificar quais pares eram significativos. Com $\alpha = 0.05$ o valor crítico para comparar o *ranking*-médio de dois algoritmos é 3.77. Na Tabela 5.26 é apresentado o *ranking* médio usando o teste de Friedman para as taxas de erro de validação e de teste. Portanto, o teste confirmou que o ACO foi significativamente melhor do que a abordagem Padrão. Além disso, a abordagem Aleatória também foi significativamente melhor do que a abordagem Padrão.

Tabela 5.26: *Ranking* Médio usando o teste de Friedman.

Algoritmo	Validação	Teste
ACO	1.000	5.125
AG	4.625	2.875
CSA	4.250	3.375
PSO	3.125	2.250
Aleatória	2.000	3.000
Padrão	6.000	4.375

A partir dos resultados armazenados, é possível obter algumas medidas considerando todas as combinações de valores para os parâmetros. Na Tabela 5.27 são apresentadas as medidas mediana, mínimo e máximo para cada conjunto de dados. Esses valores são referentes às taxas de erro de todas as combinações de valores dos parâmetros para todas as partições. Por meio dessa tabela, é possível observar que os algoritmos bioinspirados e a abordagem Aleatória conseguiram taxas de erro de validação muito próximas do mínimo encontrado para os valores de parâmetros testados. A partir dos valores da mediana é possível observar que há um grande conjunto de valores de parâmetros que produzem classificadores com pequenas taxas de erro para esses conjuntos de dados. Isso pode ser observado também pelos gráficos do Apêndice A.

Na Tabela 5.28 são apresentados os valores dos parâmetros para as RNs encontrados pelos algoritmos bioinspirados que obtiveram as menores taxa de erro de validação. Os dados apresentados são as médias dos valores encontrados por todos os algoritmos, para todas as partições e execuções. O número de neurônios utilizados pela abordagem Padrão para os conjuntos de dados Cólon, Glioma, Leucemia e Pâncreas é 9, 17, 26 e 26, respectivamente. O parâmetro η é igual a 0.3 e o parâmetro μ é igual a 0.2 para todos os conjuntos de dados. Em geral, o número de neurônios na camada oculta foi pequeno, como pode ser observado pela mediana, com exceção do conjunto Glioma, em que a me-

Tabela 5.27: Medidas da taxa de erro de cada conjunto de dados (média para as 10 partições).

Medida	Val./Teste	Cólon	Glioma	Leucemia	Pâncreas
Máximo	Validação	36.4	55.6	50.6	47.0
	Teste	42.9	56.7	52.4	48.3
Mediana	Validação	19.2	16.9	18.8	17.7
	Teste	15.1	15.0	23.0	17.3
Mínimo	Validação	12.5	11.9	12.6	12.1
	Teste	8.1	7.7	7.5	7.6

diana do número de neurônios é 81. Para a taxa de aprendizado, os valores encontrados abrangem todo o espaço de busca, com mediana de aproximadamente 0.5, para todos os conjuntos de dados. Para o parâmetro termo *momentum*, valores superiores a 0.6 foram mais usados. Porém, os valores de mínimo e máximo mostram que o espaço de busca foi, praticamente, todo explorado.

Tabela 5.28: Estatísticas dos melhores valores de parâmetros encontrados por todos os algoritmos bioinspirados.

Parâmetro	Medida	Cólon	Glioma	Leucemia	Pâncreas
γ (nr. neurônios)	Máximo	100	100	100	99
	Mediana	41	81	29	29
	Mínimo	2	3	2	2
η	Máximo	1.00	1.00	1.00	1.00
	Mediana	0.59	0.55	0.55	0.53
	Mínimo	0.05	0.05	0.05	0.05
μ	Máximo	1.00	1.00	1.00	1.00
	Mediana	0.80	0.60	0.88	0.88
	Mínimo	0.00	0.00	0.00	0.05

A taxa de erro de classificação foi a medida usada para apresentar os resultados obtidos e para a aplicação dos testes estatísticos, pois os algoritmos bioinspirados guiaram suas buscas usando a taxa de erro de validação. Neste método experimental, o desempenho dos classificadores também são avaliados pela área abaixo da curva ROC (AUC). Na Tabela 5.29 são apresentadas as áreas abaixo das curvas ROC para todos os algoritmos bioinspirados e conjuntos de dados utilizados. Entre parênteses também são apresentados os desvios padrão para as 10 partições. As maiores áreas estão destacadas. As áreas foram obtidas por classificadores gerados com os valores de parâmetros que resultaram na taxa de erro mais próxima da taxa de erro médio.

Para os conjuntos Glioma e Leucemia, o CSA obteve a maior área. No conjunto Glioma, o AG e a abordagem Padrão também conseguiram a mesma área que o CSA. Para o conjunto Cólon, o AG conseguiu uma área superior à dos outros algoritmos e abordagens. Por fim, para o conjunto Pâncreas, a abordagem Padrão conseguiu a maior área abaixo

da curva ROC. Esses resultados diferem um pouco do que foi obtido para as taxas de erro, apresentadas nas tabelas anteriores. A maior diferença ocorreu para o conjunto Glioma, pois o CSA obteve um dos piores resultados em relação à taxa de erro. As diferenças encontradas ocorreram, principalmente, por três motivos: i) devido a aleatoriedade na inicialização dos pesos das RNs, o mesmo conjunto de treinamento e parâmetros pode gerar diferentes classificadores; ii) como as diferenças entre as taxas de erro de teste não foram significativamente diferentes, podem ocorrer variações na comparação com AUC; iii) ao contrário da taxa de erro, a AUC não é sensível ao desbalanceamento de classes. O desvio padrão (calculado para as 10 partições de teste) é muito alto, principalmente para o conjunto de dados Pâncreas. Os valores de desvio padrão para a abordagem Aleatória foram, em geral, maiores do que os obtidos pelos algoritmos bioinspirados, o que significa uma maior instabilidade da rede. No Apêndice B, os gráficos das curvas ROC para cada conjunto de dados podem ser vistos nas figuras B.1 a B.4.

Tabela 5.29: Área abaixo da curva ROC para classificadores gerados utilizando valores dos parâmetros cuja taxa de erro mais se aproxima da média.

Algoritmo	Conjunto de Dados			
	Cólon	Glioma	Leucemia	Pâncreas
ACO	0.896 (0.131)	0.872 (0.210)	0.852 (0.156)	0.747 (0.213)
AG	0.938 (0.121)	0.883 (0.158)	0.920 (0.078)	0.892 (0.169)
CSA	0.927 (0.122)	0.883 (0.158)	0.924 (0.093)	0.808 (0.219)
PSO	0.908 (0.121)	0.872 (0.178)	0.910 (0.089)	0.883 (0.168)
Aleatória	0.908 (0.130)	0.822 (0.206)	0.826 (0.189)	0.833 (0.233)
Padrão	0.888 (0.171)	0.883 (0.158)	0.912 (0.129)	0.939 (0.105)

5.7.2 Resultados do Ajuste de Parâmetros de SVMs com o Método Experimental C

Nas Tabelas de 5.30 a 5.39 são apresentadas as taxas de erro (em porcentagem) de classificação para os conjuntos de validação e de teste usando os valores de parâmetros encontrados por cada algoritmo bioinspirado para as SVMs. Os valores entre parênteses são os desvios padrão para as 30 execuções dos algoritmos bioinspirados. Além disso, é apresentado o número de soluções diferentes testadas para os parâmetros. Os menores erros estão destacados.

Os resultados apresentados nas tabelas permitem observar que o PSO conseguiu os melhores resultados entre os algoritmos bioinspirados, e resultados similares à abordagem Aleatória. É possível indicar dois fatores principais que levaram ao bom desempenho da abordagem Aleatória. Primeiro, devido ao pequeno número de soluções, a abordagem Aleatória, com um pequeno número de diferentes soluções analisadas (em média 88) cobriu uma grande região do espaço de busca. O segundo é o fato de que não é difícil

Tabela 5.30: Cólón - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Cólón.

Algoritmo	Validação	Teste	# Comb.
ACO	12.3 (0.8)	14.3 (2.1)	56
AG	13.8 (0.9)	16.0 (2.8)	11
CSA	12.3 (0.3)	15.1 (1.9)	66
PSO	11.8 (0.2)	14.1 (2.4)	52
Aleatória	11.8 (0.2)	13.6 (1.7)	88
Padrão	13.7 (-)	14.3 (-)	◇

Tabela 5.31: Glioma - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Glioma.

Algoritmo	Validação	Teste	# Comb.
ACO	11.5 (0.6)	16.2 (2.2)	56
AG	13.0 (0.5)	14.9 (2.2)	11
CSA	11.2 (0.3)	17.2 (2.0)	66
PSO	11.3 (0.5)	16.9 (2.1)	54
Aleatória	10.9 (0.2)	18.0 (1.2)	88
Padrão	15.1 (-)	13.3 (-)	◇

Tabela 5.32: Leucemia - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Leucemia.

Algoritmo	Validação	Teste	# Comb.
ACO	10.4 (1)	13.5 (1.3)	56
AG	11.1 (0.9)	13.1 (1.4)	11
CSA	10.0 (0.2)	12.9 (0.9)	67
PSO	10.0 (0.1)	13.2 (0.8)	53
Aleatória	9.9 (0.1)	13.2 (1.0)	88
Padrão	11.4 (-)	12.0 (-)	◇

Tabela 5.33: Pâncreas - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Pâncreas.

Algoritmo	Validação	Teste	# Comb.
ACO	12.9 (0.9)	18.0 (1.1)	56
AG	14.1 (1.0)	17.9 (2.1)	12
CSA	12.5 (0.3)	17.7 (1.3)	68
PSO	12.2 (0.3)	17.7 (1.2)	52
Aleatória	12.1 (0.2)	17.3 (1.0)	88
Padrão	13.2 (-)	13.2 (-)	◇

Tabela 5.34: Leucemia2 - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Leucemia2.

Algoritmo	Validação	Teste	‡ Comb.
ACO	28.1 (2.0)	28.3 (2.3)	56
AG	29.1 (1.1)	27.9 (1.2)	11
CSA	27.8 (0.3)	27.2 (1.0)	65
PSO	27.1 (0.3)	27.2 (0.5)	51
Aleatória	27.2 (0.2)	27.9 (0.7)	88
Padrão	33.3 (-)	28.9 (-)	◇

Tabela 5.35: Pulmão - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Pulmão.

Algoritmo	Validação	Teste	‡ Comb.
ACO	29.8 (0.7)	37.1 (2.0)	56
AG	31.5 (1.1)	37.7 (2.6)	11
CSA	29.8 (0.3)	38.2 (1.7)	66
PSO	29.4 (0.2)	37.3 (1.5)	52
Aleatória	29.4 (0.2)	37.5 (1.2)	88
Padrão	34.0 (-)	37.2 (-)	◇

Tabela 5.36: Australian - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Australian.

Algoritmo	Validação	Teste	‡ Comb.
ACO	13.3 (0.1)	15.5 (0.8)	59
AG	13.8 (0.3)	15.0 (0.9)	11
CSA	13.3 (0.1)	15.0 (0.6)	62
PSO	13.2 (0.1)	15.2 (0.7)	55
Aleatória	13.1 (0.1)	14.8 (0.8)	88
Padrão	14.3 (-)	15.1 (-)	◇

Tabela 5.37: Pima-indians-diabetes - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Pima-indians-diabetes.

Algoritmo	Validação	Teste	‡ Comb.
ACO	22.5 (0.2)	22.5 (0.5)	57
AG	22.9 (0.2)	22.8 (0.4)	11
CSA	22.5 (0.1)	22.7 (0.5)	63
PSO	22.3 (0.1)	22.4 (0.3)	52
Aleatória	22.3 (0.0)	22.5 (0.4)	88
Padrão	24.2 (-)	24.7 (-)	◇

Tabela 5.38: Segment - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Segment.

Algoritmo	Validação	Teste	# Comb.
ACO	3.3 (1.7)	3.2 (1.7)	55
AG	4.2 (1.8)	4.0 (1.6)	10
CSA	3.2 (0.5)	3.1 (0.6)	60
PSO	2.5 (0.0)	2.5 (0.1)	57
Aleatória	2.6 (0.0)	2.6 (0.1)	88
Padrão	41.9 (-)	38.8 (-)	◇

Tabela 5.39: Vehicle - taxa de erro de validação e de teste (em %), desvio padrão e número de diferentes combinações testadas para o conjunto de dados Vehicle.

Algoritmo	Validação	Teste	# Comb.
ACO	14.5 (0.3)	16.4 (0.5)	55
AG	16.1 (0.9)	17.1 (0.8)	11
CSA	15.6 (0.5)	16.9 (0.7)	60
PSO	14.0 (0.1)	16.1 (0.3)	51
Aleatória	14.3 (0.1)	16.1 (0.4)	88
Padrão	23.0 (-)	23.7 (-)	◇

encontrar bons valores para os parâmetros da técnica SVM para os conjuntos de dados selecionados. Esse fato pode ser confirmado pelos gráficos de A.5 a A.14, exibidos no Apêndice A. Esses gráficos mostram que há uma grande densidade no início das curvas, ou seja, muitas combinações de valores para os parâmetros resultam em pequenas taxas de erro de validação e de teste.

Ao contrário do que aconteceu nos experimentos com as RNs, neste método experimental, o algoritmo ACO não obteve os melhores resultados para SVMs. O principal motivo para isso é o número reduzido de iterações (20) e indivíduos (5) utilizados. Isso acabou prejudicando o ACO, que converge lentamente. A convergência lenta pode representar uma vantagem ou uma desvantagem, dependendo do problema em que o algoritmo será aplicado. Uma convergência muito rápida pode fazer com que o algoritmo fique preso a um mínimo local, mas pode levar a uma boa solução com poucas iterações. Uma convergência mais lenta, por outro lado, pode contribuir para que o algoritmo não fique preso a mínimos locais, mas necessita de muitas iterações.

Os valores padrão utilizados para as SVMs mostraram-se robustos. Apesar dos algoritmos bioinspirados e da abordagem Aleatória terem conseguido, em geral, resultados melhores para os conjuntos de validação, a abordagem Padrão obteve os melhores resultados para as taxas de erro de teste para os conjuntos de dados Glioma, Leucemia e Pâncreas. Para os conjuntos Cólon, Leucemia2, Pulmão, *pima*, e *australian* os resultados são similares aos algoritmos bioinspirados e a abordagem Padrão. Porém, para os conjuntos de dados *segment* e *vehicle*, a abordagem Padrão obteve taxas de erro maiores,

principalmente para o conjunto *segment*, em que a diferença foi maior que 35%. Com isso, conclui-se que o ajuste de parâmetros pode trazer grandes reduções das taxas de erro para alguns conjuntos de dados. Para os conjuntos testados nestes experimentos, não foi necessário utilizar algoritmos ou técnicas sofisticadas para conseguir encontrar bons valores para os parâmetros.

Como dito anteriormente, o número de diferentes combinações testadas por cada algoritmo provê informação a respeito do compromisso entre exploração e prospecção. Assim como nos experimentos anteriores, a proporção de diferentes combinações testadas pelo ACO, PSO e CSA indica um bom balanço entre exploração e prospecção do espaço de busca. O AG explora uma proporção muito menor de combinações. Porém, os resultados do AG são comparáveis aos obtidos por outros algoritmos. Novamente isso pode ser atribuído ao fato de não ser difícil encontrar bons valores para os parâmetros da técnica SVM para esses conjuntos de dados.

Assim como nos outros experimentos, foram realizados testes estatísticos para avaliar as diferenças observadas. Para isso, foram usadas as taxas de erro de validação e de teste obtidas pelos classificadores. Primeiramente, foi aplicado o teste-*t* pareado corrigido, com nível de significância $\alpha = 0.05$ (95% de confiança) e nove graus de liberdade, correspondente às $k - 1$ partições. A hipótese nula é que os algoritmos e as abordagens conseguiram o mesmo desempenho. A hipótese nula foi rejeitada para algumas comparações, ou seja, o teste estatístico mostrou que há algumas diferenças significativas para as taxas de erro de teste. Os resultados para o teste estatístico realizado são apresentados na Tabela 5.40. Cada célula $C_{i,j}$ representa a comparação entre o algoritmo/abordagem da linha i com o algoritmo/abordagem da coluna j . Cada célula possui uma letra indicando o conjunto de dados em que o algoritmo/abordagem i é significativamente diferente do algoritmo/abordagem j : {C}olon, {G}lioma, {L}eucemia, {P}âncreas, L{E}ucemia2, P{U}lmão, {A}ustralian, p{I}ma, {S}egment, {V}ehicle. Um hífen é usado quando a diferença não é significativa.

Em geral, o teste confirmou os resultados comentados anteriormente. Todos os algoritmos bioinspirados e a abordagem Aleatória obtiveram taxas de erro de teste significativamente melhores do que a abordagem Padrão para os conjuntos de dados *segment* e *vehicle*. Para o conjunto *segment*, a abordagem Aleatória e o algoritmo PSO foram significativamente diferentes dos algoritmos AG e CSA. O teste ainda mostrou que os algoritmos bioinspirados foram significativamente diferentes da abordagem Padrão para o conjunto *pima*. Por fim, o algoritmo PSO e o ACO foram significativamente diferentes para os conjuntos de dados Leucemia e Leucemia2.

Adicionalmente aplicou-se o teste de Friedman (Demšar, 2006), para verificar se há diferença significativa entre qualquer uma das técnicas, considerando todos os conjuntos de dados. O valor crítico da distribuição F com $\alpha = 0.05$ e $5(c-1)$ e $45((c-1) \times (d-1))$ graus de liberdade é 2.42, em que c é o número de classificadores e d é o número de conjuntos

Tabela 5.40: Análise estatística - teste- t pareado corrigido (conjunto de dados de teste).

	AG	CSA	PSO	Aleatória	Padrão
ACO	-----	-----	--L-E----	-----	-----ISV
AG		-----	-----S-	-----S-	-----ISV
CSA			-----S-	-----S-	-----ISV
PSO				-----	-----ISV
Aleatória					-----SV

de dados. Como o teste resultou em $F = 1.22$, a hipótese nula, de que não há diferença estatística entre os resultados obtidos pelos algoritmos e as abordagens, foi aceita. Assim, aplicou-se o mesmo teste para o conjunto de validação, que resultou na rejeição da hipótese nula, com $F = 85.88$. Em seguida, aplicou-se o teste de Nemenyi (Demšar, 2006) para identificar quais pares eram significativos. Com $\alpha = 0.05$, o valor crítico para comparar o *ranking*-médio de dois algoritmos diferentes é 2.38. Na Tabela 5.41 é apresentado o *ranking*-médio usando o teste de Friedman para as taxas de erro de validação e de teste. Como é possível observar, o teste mostrou que a abordagem Aleatória e o algoritmo PSO são melhores que a abordagem Padrão e o algoritmo AG. Além disso, os algoritmos CSA e ACO são melhores que a abordagem Padrão. Esses resultados confirmam os resultados descritos anteriormente.

A mesma conclusão do teste- t pareado corrigido pode ser utilizada para o teste de Friedman. O ajuste de parâmetros é necessário, mesmo que não seja por um algoritmo ou técnica sofisticada. Se as estimativas para os conjuntos de dados de expressão gênica fossem tão boas quanto as obtidas para os conjuntos de dados da UCI, as diferenças observadas para as taxas de erro de validação poderiam refletir em diferenças nas taxas de erro de teste.

Tabela 5.41: *Ranking* médio usando o teste de Friedman.

Algoritmo	Validação	Teste
ACO	3.60	4.10
AG	5.20	4.20
CSA	3.30	3.65
PSO	1.60	2.65
Aleatória	1.50	2.80
Padrão	5.80	3.60

Na Tabela 5.42 são apresentadas as correlações médias, entre as taxas de erro de validação e de teste para as 10 partições utilizadas, para todas as possíveis combinações de valores para os parâmetros. É possível observar que conjuntos de dados com um número maior de exemplos possuem, em geral, uma correlação maior. Outros fatores que influenciam na correlação são a característica determinística do algoritmo de aprendizado, a dificuldade de classificação do conjunto de dados e o viés na seleção dos exemplos de

treinamento e de teste. Como pode ser observado, as correlações usando SVMs foram maiores do que as correlações obtidas pelas RNs nesse experimento (Tabela 5.24), com exceção do conjunto Glioma. Os gráficos de A.5 a A.14, exibidos no Apêndice A, mostram que a densidade das taxas de erro de validação e de teste são diferentes para conjuntos de dados com baixa correlação e parecidos para conjuntos de dados com alta correlação. Além disso, para conjuntos de dados com um grande número de exemplos, como *segment* e *vehicle*, a variação da densidade entre as partições é pequena, ao contrário do que ocorre em conjuntos de dados com poucos exemplos.

Tabela 5.42: Correlação das taxas de erro de validação e de teste para todas as combinações de parâmetros.

Conjunto de dados	Média	Desvio Padrão
Cólon	0.630	0.266
Glioma	0.681	0.275
Leucemia	0.922	0.057
Pâncreas	0.905	0.066
Leucemia2	0.959	0.048
Pulmão	0.665	0.402
<i>Australian</i>	0.972	0.017
<i>Pima</i>	0.876	0.068
<i>Segment</i>	0.998	0.000
<i>Vehicle</i>	0.987	0.003

Como os resultados para todos os possíveis valores de parâmetros estão armazenados em uma estrutura de dados, é possível obter algumas medidas. Na Tabela 5.43 são apresentadas a mediana, o mínimo e o máximo para as taxas de erro de cada conjunto de dados. Esses valores são as médias para todas as partições de todas as combinações de valores dos parâmetros. Por meio desta tabela, é possível observar que os algoritmos bioinspirados e a abordagem Aleatória conseguiram taxas de erro de validação mínimas ou muito próximas disso. Isso novamente mostra que há um grande conjunto de valores de parâmetros que produzem classificadores com baixa taxa de erro de classificação para esses conjuntos de dados.

Na Tabela 5.44 são apresentadas algumas medidas referentes aos valores de parâmetros encontrados pelos algoritmos bioinspirados. Os dados apresentados são as médias dos valores encontrados em todas as partições e execuções. Na última coluna do parâmetro γ são apresentados os valores usados pela abordagem Padrão para esse parâmetro. O valor do parâmetro C é sempre 1, portanto não está na tabela. Os valores encontrados para o parâmetro custo (C) abrangem quase todo o espaço de busca. Porém, nenhum algoritmo encontrou valores próximos à extremidade inferior (-5) para esse parâmetro, ao contrário da extremidade superior (15). Para o parâmetro *gamma* (γ), o valor máximo médio não é maior do que -0.5 , sendo que o limite superior é 3 . O valor mínimo médio foi usado para quase todos os conjuntos de dados. Com esses resultados, não é possível definir um

Tabela 5.43: Medidas da taxa de erro de cada conjunto de dados (média para as 10 partições).

	Mínimo		Máximo		Mediana	
	Validação	Teste	Validação	Teste	Validação	Teste
Cólon	11.7	3.4	36.7	39.2	26.8	26.7
Glioma	10.3	6.3	45.7	50.0	22.5	23.3
Leucemia	9.6	5.6	49.1	50.6	27.9	24.4
Pâncreas	11.7	7.2	47.3	46.7	21.8	26.5
Leucemia2	26.0	20.2	68.1	68.0	62.5	62.5
Pulmão	29.3	20.4	57.6	61.8	55.8	57.0
<i>Australian</i>	13.0	9.6	45.0	45.2	19.1	19.0
<i>Pima</i>	22.2	18.5	35.5	37.9	29.9	28.6
<i>Segment</i>	2.5	1.6	72.8	71.4	24.7	23.4
<i>Vehicle</i>	14.0	11.5	74.2	74.3	28.4	28.4

intervalo muito pequeno para busca dos parâmetros. Porém, o valor de γ não precisa ser maior que 1 e o valor de C não precisa ser menor que 2^{-5} .

Tabela 5.44: Estatísticas dos melhores valores de parâmetros encontrados por todos os algoritmos bioinspirados (2^v , em que v é o valor encontrado para o parâmetro).

Algoritmo	Custo (C)			Gama (γ)			
	Mediana	Min.	Max.	Mediana	Min.	Max.	Padrão
Cólon	4.8	-1.2	15.0	-9.0	-15.0	-2.0	-5.0
Glioma	4.8	-2.0	15.0	-2.2	-15.0	-0.5	-4.0
Leucemia	3.0	-3.0	15.0	-5.2	-15.0	-1.0	-6.0
Pâncreas	3.5	-2.2	15.0	-6.0	-15.0	-1.0	-6.0
Leucemia2	5.5	-1.0	15.0	-8.0	-15.0	-2.0	-6.0
Pulmão	7.2	-0.5	15.0	-8.0	-14.8	-2.0	-6.0
<i>Australian</i>	1.5	-4.2	14.8	-7.8	-15.0	-2.0	-4.0
<i>Pima</i>	3.0	-3.0	15.0	-8.8	-15.0	-1.8	-3.0
<i>Segment</i>	7.8	-1.0	14.8	-12.2	-15.0	-4.0	-4.0
<i>Vehicle</i>	10.2	1.2	15.0	-6.6	-12.2	-2.5	-4.0

Assim como para as RNs, na Tabela 5.45 são apresentados os valores da AUC para os conjuntos de teste. Isso é feito para todos os algoritmos bioinspirados e conjuntos de dados binários. No Apêndice B, os gráficos das curvas ROC para cada conjunto de dados podem ser vistos nas figuras B.5 a B.10. Os gráficos das curvas ROC são apresentados para cada conjunto de dados. O mesmo procedimento utilizado para as RNs foi adotado para as SVMs. Essas áreas foram obtidas por classificadores gerados com os valores de parâmetros que resultaram na taxa de erro mais próxima da taxa de erro médio.

É possível observar que, em geral, para os conjuntos de dados com maior número de exemplos, como o *australian* e o *pima*, os desvios padrão são menores do que conjuntos de dados com um número pequeno de exemplos. Os desvios padrão da AUC para os

conjuntos Cólon, Glioma, Leucemia e Pâncreas foram menores do que os desvios padrão obtidos pelas RNs. Algumas diferenças foram observadas em relação às taxas de erro apresentadas anteriormente. A abordagem Padrão não obteve uma área grande para o conjunto Glioma. A área obtida pelo ACO para o Pâncreas foi igual à área obtida pela abordagem Padrão. Em geral, o algoritmo PSO conseguiu bons resultados para todos os conjuntos de dados.

Tabela 5.45: Área abaixo da curva ROC para classificadores gerados utilizando valores dos parâmetros cuja taxa de erro mais se aproxima da média.

Algoritmo	Conjunto de Dados					
	Cólon	Glioma	Leucemia	Pâncreas	<i>Australian</i>	<i>Pima</i>
ACO	0.938 (0.088)	0.900 (0.119)	0.932 (0.098)	0.958 (0.090)	0.922 (0.027)	0.836 (0.053)
AG	0.863 (0.171)	0.956 (0.073)	0.952 (0.073)	0.922 (0.108)	0.918 (0.029)	0.825 (0.052)
CSA	0.913 (0.103)	0.922 (0.084)	0.936 (0.098)	0.922 (0.108)	0.918 (0.029)	0.828 (0.056)
PSO	0.925 (0.105)	0.939 (0.081)	0.960 (0.065)	0.958 (0.090)	0.932 (0.022)	0.827 (0.055)
Aleatória	0.938 (0.088)	0.922 (0.115)	0.908 (0.103)	0.958 (0.090)	0.926 (0.028)	0.831 (0.051)
Padrão	0.871 (0.132)	0.922 (0.115)	0.964 (0.048)	0.958 (0.090)	0.920 (0.034)	0.822 (0.052)

5.7.3 Comparação entre RNs e SVMs no Domínio de Expressão Gênica

Nesta seção, as técnicas RNs e SVMs são brevemente comparadas quando aplicadas para classificação de dados de expressão gênica. Os quatro conjuntos de dados de expressão gênica (Cólon, Glioma, Pâncreas e Leucemia) classificados pelas duas técnicas são utilizados na comparação. Os conjuntos de dados Leucemia2 e Pulmão não foram aplicados às RNs, devido ao tempo de processamento necessário. Portanto, não puderam ser usados na comparação. Na Tabela 5.46 são apresentadas algumas medidas para comparação dos classificadores. Nas linhas nomeadas *Diferença*, é computada a diferença entre os resultados obtidos para as RNs e SVMs. A primeira comparação é entre as taxas de erro obtidas pelos classificadores gerados com o ajuste de parâmetros. Na comparação, foram usados os melhores resultados obtidos para os conjuntos de dados mencionados (tabelas 5.16-5.19 para RNs e tabelas 5.30- 5.33 para SVMs). Em seguida, são comparadas as taxas de erro de teste mínimas entre todos as combinações de valores para os parâmetros definidos. Os valores foram retirados das tabelas 5.27 e 5.43. Por último são comparadas as áreas abaixo da curva ROC. Os valores foram retirados das tabelas 5.29 e 5.45, para RNs e SVMs, respectivamente.

Como é possível observar, as SVMs conseguiram resultados melhores que as RNs para todas as comparações feitas. Houve apenas um empate na comparação da AUC para o conjunto Cólon. Uma importante observação é que as SVMs conseguiram taxas de erro de teste menores do que as RNs. De fato, SVMs representam, até o momento, a técnica de AM com melhores resultados na classificação de dados de expressão gênica (Statnikov

Tabela 5.46: Comparação entre RNs e SVMs para classificação de dados de expressão gênica.

Medida	Conjunto de Dados			
	Cólon	Glioma	Leucemia	Pâncreas
Ajuste RNs	15.8	14.0	19.2	17.3
Ajuste SVMs	13.6	13.3	12.0	13.2
Diferença	2.2	0.7	7.2	4.1
Erro mínimo RNs	8.1	7.7	7.5	7.6
Erro mínimo SVMs	3.4	6.3	5.6	7.2
Diferença	4.7	1.4	1.9	0.4
AUC RNs	0.938	0.883	0.924	0.939
AUC SVMs	0.938	0.956	0.964	0.958
Diferença	0.000	-0.073	-0.040	-0.019

et al., 2008). Diaz-Uriarte & Alvarez de Andres (2006) utilizaram Florestas Aleatórias (RF, do inglês, *Random Forests*) (Breiman, 2001) para selecionar e classificar dados de *microarray*. Nesse trabalho, os autores obtiveram resultados que sugeriam que RFs possuem desempenho comparável às SVMs para classificação de dados de expressão gênica. Porém, Statnikov et al. (2008) apresentaram vários viés que poderiam ter levado a esses resultados:

- As RFs foram aplicadas aos conjuntos de dados antes da seleção de genes, enquanto que as SVMs foram aplicadas a uma seleção de 200 genes (esse número foi escolhido arbitrariamente);
- A estratégia um-contra-um foi utilizada para geração das SVMs multiclases. Porém, sabe-se que outras estratégias, como a um-contra-todos, são melhores para dados de expressão gênica (Statnikov et al., 2005a; Rifkin et al., 2003);
- Foram utilizadas apenas SVMs lineares, sem ajuste de parâmetros;
- A métrica de avaliação usada (proporção de classificações corretas) é sensível à distribuição desbalanceada de classes;
- Não foram realizados testes estatísticos na comparação dos classificadores;
- Para a estimação do erro foi usado o método .632, que não é o mais apropriado para dados de *microarray*.

A partir da hipótese de que esses viés comprometeram os resultados obtidos por Diaz-Uriarte & Alvarez de Andres (2006), Statnikov et al. (2008) realizaram uma comparação metodologicamente mais rigorosa entre os dois algoritmos para uma extensa variedade de conjuntos de dados de expressão gênica. Foi verificado que a hipótese é verdadeira, ou seja, as limitações apontadas podem ter comprometido a comparação entre RFs e SVMs.

Após eliminar essas limitações, observou-se que as RFs apresentaram, para a maioria e na média, erros de classificação maiores do que as SVMs, quando a seleção de genes é realizada e também quando a seleção não é realizada. A comparação entre RNs e SVMs realizada nesta dissertação não pode ser interpretada como conclusiva, pois também apresenta viés de seleção de genes.

Além do melhor desempenho em termos de taxas de erro e AUC, a criação das estruturas de dados com os resultados das classificações foi mais rápida para as SVMs. O principal motivo para a diferença no tempo de processamento foi a maior quantidade de combinações de parâmetros para as RNs e as 10 repetições necessárias para cada treinamento da RN, já que os pesos iniciais das conexões são definidos aleatoriamente, sendo necessárias repetições para obter uma estimativa razoável. Adicionalmente, um grande número de neurônios na camada oculta torna o aprendizado mais lento.

5.8 Considerações Finais

Neste capítulo foram apresentados os experimentos realizados para investigar o ajuste de parâmetros de SVMs e RNs utilizando algoritmos bioinspirados. Foram utilizados conjuntos de dados de expressão gênica e de outros domínios. Inicialmente, foram apresentados os conjuntos de dados, os algoritmos bioinspirados, as técnicas de classificação e as métricas empregadas na avaliação dos classificadores gerados. Em seguida, foram apresentados os três métodos experimentais propostos e os resultados experimentais para cada um deles.

No método experimental A, a geração do classificador é baseada no método de validação cruzada estratificada com k partições. Uma partição é usada para teste, outra para validação e as demais para treinamento do algoritmo de aprendizado. A taxa de erro obtida para a partição de validação é usada como guia para a busca dos algoritmos bioinspirados.

No método experimental B, a geração do classificador é baseada em dois laços aninhados. Em cada um dos laços é usado o método de validação cruzada estratificada com k partições. O laço interno é usado para determinar a melhor combinação de valores para os parâmetros e o laço externo é usado para estimar o desempenho do classificador gerado com os melhores parâmetros encontrados no laço interno.

No método experimental C também são usados dois laços com validação cruzada. A diferença é que os parâmetros são ajustados para cada partição do laço externo. Essa modificação foi necessária para evitar o uso indireto dos exemplos do conjunto de teste no processo de ajuste de parâmetros.

Duas abordagens de referência foram usadas na comparação com os algoritmos bioinspirados nos três métodos propostos. Os resultados experimentais mostram que, em geral, os algoritmos bioinspirados conseguem menores taxas de erro de validação em compara-

ção com as abordagens de referência. Para o conjunto de dados em que ocorreu o maior ganho com o ajuste de parâmetros, a diferença entre a taxa de erro de teste da abordagem Padrão para o algoritmo PSO foi maior que 35%. Entretanto, para a maioria dos conjuntos de dados utilizados, a melhoria obtida nas taxas de erro de validação não é grande o suficiente para se refletir nas taxas de erro de teste. Em alguns casos, foi observado que o ajuste de parâmetros causou *overfitting*.

Capítulo 6

Conclusão

Diversos trabalhos encontrados na literatura tratam do problema de ajuste de parâmetros para SVMs (Lorena & Carvalho, 2006; Huang & Wang, 2006; Souza & Carvalho, 2005; Souza et al., 2006; Imbault & Lebart, 2004; Zhang & Jiao, 2005; Acevedo et al., 2006) e para RNs (Castillo et al., 2007; Gao et al., 2006; Braun & Weisbrod, 1993; Dodd, 1990; Leung et al., 2003; Tsai et al., 2006). Muitos deles utilizam algoritmos bioinspirados para isso. Contudo, não foram encontrados trabalhos que comparem diferentes algoritmos bioinspirados para diferentes técnicas de AM. O trabalho de (Hoste & Daelemans, 2005) é o que mais se aproxima do que foi desenvolvido nesta dissertação. Os autores utilizam AGs para ajustar parâmetros de duas técnicas de classificação (TIMBL e RIPPER) e avaliam os efeitos desse ajuste.

Neste trabalho foi investigado o uso de algoritmos bioinspirados para ajustar os valores dos parâmetros livres de duas técnicas de classificação, RNs e SVMs. Esses algoritmos foram comparados com outras abordagens usando como critério o desempenho dos classificadores gerados para dados de expressão gênica. Essa escolha teve como objetivo restringir o domínio para investigar se algum algoritmo é mais adequado nesse domínio. Para avaliar os resultados obtidos com dados de expressão gênica, conjuntos de outros domínios, obtidos do repositório UCI, foram também usados em alguns experimentos com SVMs.

Os algoritmos bioinspirados ACO, AGs, CSA e PSO foram investigados neste trabalho para o ajuste de parâmetros de SVMs e RNs. Os algoritmos foram implementados na linguagem R e as técnicas de classificação foram simuladas a partir de ferramentas computacionais disponíveis em bibliotecas que as implementam.

Foram investigados três métodos experimentais nesta dissertação. Esses métodos representam uma seqüência de investigações no sentido de melhorar as estimativas de desempenho das técnicas de classificação e evitar o uso indireto dos dados do conjunto de teste no processo de busca dos algoritmos bioinspirados. Os métodos experimentais A e B possuem viés no processo de ajuste, pois usam indiretamente dados utilizados na estimativa do erro real do classificador. No método experimental C, os parâmetros são ajustados para cada partição do método de validação cruzada, eliminando esse problema.

A seguir, na Seção 6.1, os principais resultados obtidos são apresentados. Na Seção 6.2 são discutidas as contribuições deste trabalho. As limitações, os problemas encontrados e os trabalhos futuros motivados por esses são discutidos na Seção 6.3.

6.1 Principais Resultados

Nesta seção são apresentados os principais resultados dos experimentos realizados com funções de *benchmark* e para cada método experimental de ajuste de parâmetros. No final da seção são feitas algumas discussões gerais sobre os resultados e a comparação entre RNs e SVMs para dados de expressão gênica.

Antes de serem empregados para o problema de ajuste de parâmetros, os algoritmos bioinspirados investigados foram aplicados na minimização de funções de *benchmark*. O objetivo foi avaliar a funcionalidade e o desempenho desses algoritmos para funções unimodais e multimodais. Para as duas funções unimodais testadas, Esfera e Rosenbrock, o algoritmo PSO conseguiu os melhores resultados e também convergiu rapidamente para regiões promissoras do espaço de busca. O algoritmo CSA foi o pior para a função Esfera, enquanto que o AG foi o pior para a função Rosenbrock e teve uma convergência lenta. Não foi possível determinar o melhor algoritmo para as funções multimodais, pois, para cada uma das três funções analisadas, um algoritmo diferente se destacou. Para a função Rastrigin, o AG obteve os melhores valores e o ACO os piores. O algoritmo CSA foi o melhor para a função Schaffer f6, enquanto o AG foi o pior. Por último, para a função Ackley, o ACO, apesar de convergir um pouco mais lentamente que os demais, foi o algoritmo com o melhor desempenho e o CSA o pior.

Para o ajuste de parâmetros realizado no método experimental A, foram realizados experimentos com quatro conjuntos de dados binários aplicados às SVMs com *kernel* Gaussiano. Em geral, os algoritmos bioinspirados conseguiram melhores resultados do que as abordagens de referência utilizadas. Os algoritmos conseguiram as menores taxas de erro de validação para três conjuntos de dados e as menores taxas de erro de teste para dois conjuntos. Os valores padrão utilizados para os parâmetros se mostraram robustos, apresentando taxas de erro de teste melhores para os conjuntos de dados Cólon e Glioma e resultados similares aos dos algoritmos para os conjuntos de Leucemia e Pâncreas. A abordagem Grade conseguiu resultados similares aos algoritmos bioinspirados em relação à taxa de erro de teste. Diferentes valores de parâmetros foram obtidos pelos algoritmos bioinspirados, evidenciando o grande número de mínimos locais. Com o teste de Friedman foi verificado que todos os algoritmos e abordagens tiveram desempenhos estatisticamente semelhantes.

O método experimental B foi aplicado para ajuste de parâmetros de RNs MLP. Quatro conjuntos de dados binários de expressão gênica foram usados nos experimentos. Como os resultados foram armazenados, algumas informações puderam ser extraídas, como a

correlação entre as taxas de erro de validação e de teste. Como essas taxas estavam fortemente correlacionadas, os algoritmos puderam guiar suas buscas a partir das taxas de erro de validação. Em geral, os algoritmos bioinspirados conseguiram os melhores resultados. Porém, foi observado que um grande número de valores para os parâmetros resultavam em pequenas taxas de erro, o que justifica o bom desempenho da abordagem Aleatória. O AG testou um número menor de diferentes valores para os parâmetros em relação aos demais algoritmos. Todavia, seus resultados foram comparáveis aos demais, devido ao grande número de valores que resultam em pequenas taxas de erro. De acordo com o teste de Friedman, não há diferenças significativas nos resultados obtidos para as taxas de erro de teste. Para as taxas de erro de validação, o teste mostrou que o ACO foi significativamente melhor do que a abordagem Padrão. Uma justificativa para essa diferença é que, em geral, a abordagem Padrão conseguiu menores taxas de erro de teste do que de validação, ao contrário dos algoritmos bioinspirados.

No método experimental C, o ajuste de parâmetros foi realizado para RNs MLP e SVMs com *kernel* Gaussiano. Para os experimentos com RNs, foram usados novamente os mesmos conjuntos de dados dos métodos experimentais A e B. Foi observado que as taxas de erro de validação e de teste não estavam altamente correlacionadas. Por esse motivo, ocorreram maiores variações entre essas taxas. Os algoritmos conseguiram taxas de erro de validação bem menores que a abordagem Padrão. Porém, em relação à taxa de erro de teste, foram melhores apenas para os conjuntos de dados Cólon e Glioma. Para a AUC, os desvios padrão para a abordagem Aleatória foram, em geral, maiores do que para os algoritmos bioinspirados. O teste de Friedman encontrou diferenças significativas apenas para as taxas de erro de validação. O ACO e a abordagem Aleatória foram significativamente melhores do que a abordagem Padrão.

No ajuste de parâmetros de SVMs, foram usados seis conjuntos de dados de expressão gênica e quatro conjuntos obtidos do repositório UCI. Em geral, conjuntos de dados com maior número de exemplos apresentaram correlação maior entre as taxas de erro de validação e de teste do que conjuntos com poucos exemplos. Como os conjuntos de dados de expressão gênica possuem poucos exemplos, a correlação entre as taxas de erro não foi maior que 0.7 para três deles. As correlações utilizando SVMs foram maiores do que RNs para os mesmos conjuntos de dados. Em geral, o algoritmo PSO e a abordagem Aleatória conseguiram os melhores resultados. Podem ser apontados dois fatores para o bom desempenho da abordagem Aleatória. O primeiro é o fato de não ser difícil encontrar valores de parâmetros que produzem pequenas taxas de erro de validação. Em segundo, o número reduzido de soluções que poderiam ser testadas para SVMs favorecem a abordagem Aleatória, que consegue analisar um número maior de diferentes soluções. O principal motivo para o mal desempenho do algoritmo ACO para os experimentos com SVMs é o número reduzido de iterações utilizadas, pois ele converge lentamente. Assim como para RNs, o teste de Friedman encontrou diferenças significativas apenas entre as

taxas de erro de validação. Os algoritmos PSO, CSA e ACO e a abordagem Aleatória foram significativamente melhores do que a abordagem Padrão. Além disso a abordagem Aleatória e o algoritmo PSO foram melhores do que o AG.

Os resultados experimentais mostram que, em geral, os algoritmos bioinspirados conseguem menores taxas de erro de validação em comparação com as abordagens de referência. Para a maioria dos conjuntos de dados e experimentos realizados, a melhoria obtida nas taxas de erro de validação não foi grande o suficiente para resultar em melhoria nas taxas de erro de teste. O maior ganho com o ajuste de parâmetros ocorreu para o conjunto de dados *segment*, que possui o maior número de exemplos e classes, nos experimentos com SVMs. A diferença entre a taxa de erro de teste da abordagem Padrão para o algoritmo PSO foi maior que 35%. Uma diferença acima de 7.5% também foi obtida no conjunto de dados *vehicle*, o segundo maior conjunto de dados, com quatro classes. Os valores padrão para as SVMs e RNs obtiveram bons resultados para a maior parte dos conjuntos de dados. Quando isso não ocorreu, encontrar valores que resultam em melhor desempenho não foi uma tarefa difícil, pois há uma grande quantidade de valores que conseguem minimizar a taxa de erro de validação. Em alguns casos, foi observado que o ajuste de parâmetros causou *overfitting*.

Na última seção dos experimentos para ajuste de parâmetros, foram realizadas algumas comparações entre RNs e SVMs para classificação de dados de expressão gênica. Primeiramente, as técnicas foram comparadas em relação às melhores taxas de erro de teste obtidas com o processo de ajuste de parâmetros. As SVMs conseguiram os melhores resultados para os quatro conjuntos de dados utilizados. A menor diferença entre as taxas de erro foi de 0.7%, para o conjunto Glioma, e a maior de 7.2%, para o conjunto Leucemia. Entre todos os valores de parâmetros definidos para as duas técnicas, as SVMs novamente obtiveram as menores taxas de erro de teste para os quatro conjuntos de dados, com uma diferença mínima de 0.4% e máxima de 4.7% em relação às RNs. Devido aos viés descritos na Seção 5.7.3, não é possível obter conclusões com a comparação realizada. Porém, os resultados apontam para um melhor desempenho das SVMs.

6.2 Contribuições

As principais contribuições deste trabalho são descritas a seguir:

- Análise da sensibilidade das SVMs e RNs em relação aos valores dos parâmetros para dados de expressão gênica. É conhecido que SVMs e RNs são sensíveis aos valores dos parâmetros utilizados. Este estudo avalia a sensibilidade dessas técnicas restrito ao domínio de expressão gênica;
- Comparação entre os algoritmos bioinspirados ACO, AGs, CSA e PSO para o ajuste de parâmetros de SVMs e RNs. Na literatura, há inúmeros trabalhos que usam algo-

ritmos bioinspirados para o ajuste de parâmetros de técnicas de classificação. Porém, não foram encontrados trabalhos que comparem diferentes algoritmos bioinspirados para ajuste de uma técnica de AM. Este trabalho realiza essa comparação para dados de expressão gênica utilizando RNs e também para outros domínios com SVMs;

- Comparação entre RNs e SVMs para classificação de dados de expressão gênica. O problema de viés de seleção de genes nos experimentos realizados é reconhecido. Entretanto, este estudo fornece indícios do desempenho das duas técnicas de classificação para dados de expressão gênica.
- Desenvolvimento de um conjunto de ferramentas computacionais para o ajuste de parâmetros de técnicas de classificação. Os algoritmos bioinspirados investigados e os métodos experimentais para ajuste de parâmetros foram implementados na linguagem R. Essas ferramentas podem ser utilizadas para realizar o ajuste de parâmetros de outras técnicas de classificação. Os dados resultantes dos experimentos ficam armazenados em estruturas de dados e podem ser usados em futuras investigações.

Algumas das contribuições citadas foram publicadas em congressos nacionais e internacionais (Rossi & Carvalho, 2008; Rossi et al., 2008a,b). Essas contribuições resumem alguns experimentos realizados durante os dois anos de mestrado.

6.3 Limitações, Problemas Encontrados e Trabalhos Futuros

Nesta seção são descritas e discutidas as limitações deste trabalho, os problemas encontrados durante o seu desenvolvimento e os trabalhos futuros, motivados pelos problemas e limitações encontrados.

A seleção de atributos em análise de dados de expressão gênica é importante não apenas para reduzir o custo computacional, pois normalmente esses conjuntos de dados possuem milhares de genes, mas também eliminar genes não informativos. Neste trabalho, o problema de viés de seleção de atributos é reconhecido. Entretanto, decidiu-se não tratá-lo, pois como o objetivo não é encontrar o melhor classificador, mas verificar o desempenho de diferentes algoritmos bioinspirados no ajuste de parâmetros, o trabalho não sofre grande influência da seleção de genes. Em trabalhos futuros, essa seleção de genes pode ser realizada simultaneamente com o ajuste de parâmetros.

O uso de uma estrutura de dados para armazenar os resultados dos classificadores economiza tempo de processamento, pois não é necessário executar o algoritmo de aprendizado milhares de vezes para cada algoritmo bioinspirado. O armazenamento dos resultados também viabiliza a extração de algumas informações, como a correlação, que podem

auxiliar na análise desses resultados e na tomada de decisões do estudo. Porém, para que o armazenamento seja possível, os intervalos de valores para cada parâmetro precisam ser discretizados. Durante a discretização, podem ser perdidos valores que poderiam resultar em menores taxas de erro ou até mesmo em soluções ótimas. Dessa maneira, seria interessante realizar experimentos preliminares com os intervalos de valores definidos para os parâmetros discretizados, e, posteriormente, se os resultados se mostrarem promissores, realizar uma busca contínua para esses intervalos.

Segundo Lourenço et al. (2002) e Ide & Yasuda (2005), os algoritmos bioinspirados são robustos. Todavia, os valores para seus parâmetros podem também influenciar em seus desempenhos. Nesta pesquisa, os valores para esses algoritmos foram definidos com base em trabalhos previamente publicados e em experimentos com funções de *benchmark*. O ajuste desses parâmetros poderia implicar em diferentes resultados. Porém, seria inviável incluir mais esse estudo neste trabalho.

Para todos os experimentos com RNs, foram utilizados apenas quatro conjuntos de dados. Esse número não é o ideal, mas devido ao alto custo computacional com os experimentos para RNs, não foi possível utilizar mais conjuntos. Os principais fatores para o alto custo computacional nos experimentos com RNs se devem às repetições necessárias para obter boas estimativas e ao treinamento lento quando é usado um grande número de neurônios. Outra limitação é o uso de apenas uma camada oculta e da arquitetura completamente conectada. Assim, seria interessante pesquisar um meio para reduzir o tempo computacional, testar redes com mais de uma camada e possibilitar a remoção de conexões, ou seja, utilizar redes parcialmente conectadas, o que poderia acelerar o processo de treinamento e melhorar o desempenho.

Nos experimentos com SVMs, foi usado apenas o *kernel* Gaussiano, que, segundo Hsu et al. (2007), apresenta, em geral, melhores resultados e menos dificuldades numéricas. Porém, alguns trabalhos utilizam o *kernel* polinomial, como em Statnikov et al. (2008) e Statnikov et al. (2005a). Portanto, além do ajuste de parâmetros, seria interessante testar outros *kernels* para avaliar se os resultados são alterados.

Referências Bibliográficas

- Abbas, A., Lichtman, A., & Pober, J. (2000). *Cellular and Molecular Immunology*. W.B. Saunders, New York, 4 edition. (Citado na página 30.)
- Acevedo, J., Maldonado-Bascón, S., Lafuente-Arroyo, S., Gómez-Moreno, H., & Gil-Jiménez, P. (2006). Model selection for support vector machines using ant colony optimization in an electronic nose application. In Dorigo, M., Gambardella, L. M., Birattari, M., Martinoli, A., Poli, R., & Stützle, T., editors, *ANTS Workshop*, volume 4150 of *Lecture Notes in Computer Science*, pag. 468–475. Springer. (Citado na página 85.)
- Achcar, J. A. & Rodrigues, J. (2000). Introdução à estatística para ciências e tecnologia. Technical report, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. (Citado na página 39.)
- Ackley, D. H. (1987). *A Connectionist Machine for Genetic Hillclimbing*. Kluwer Academic Publishers, Boston, USA. (Citado na página 36.)
- Aimo, T. & Zilinskas, A. (1989). *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer-Verlag. (Citado na página 36.)
- Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723. (Citado na página 18.)
- Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750. (Citado na página 46.)
- Alpaydin, E. (2004). *Introduction to Machine Learning: Adaptive Computation and Machine Learning*. The MIT Press. (Citado na página 7.)
- Anders, U. & Korn, O. (1999). Model selection in neural networks. *Neural Networks*, 12:309–323. (Citado na página 18.)
- Asuncion, A. & Newman, D. J. (2007). UCI machine learning repository. <http://www.ics.uci.edu/~mlern/MLRepository.html>. (Citado nas páginas 3, 6 e 45.)

- Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK. (Citado na página 36.)
- Bajcsy, P., Jiawei Han, L. L., & Yang, J. (2005). Survey of biodata analysis from a data mining perspective. In Wang, J. T. L., Zaki, M. J., Toivonen, H. T. T., & Shasha, D., editors, *Data Mining in Bioinformatics*, capítulo 2, pag. 9–38. Springer-Verlag. (Citado nas páginas 9 e 10.)
- Bartlett, P. L., Boucheron, S., & Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*, 48:85–113. (Citado na página 18.)
- Basheer, I. A. & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43:3–31. (Citado nas páginas 2 e 17.)
- Beer, D. G., Kardia, S. L., Huang, C.-C., Giordano, T. J., Levin, A. M., Misek, D. E., Lin, L., Chen, G., Gharib, T. G., Thomas, D. G., Lizyness, M. L., Kuick, R., Hayasaka, S., Taylor, J. M., Iannettoni, M. D., Orringer, M. B., & Hanash, S. (2002). Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 8:816–824. (Citado na página 46.)
- Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., & Yakhini, Z. (2000). Tissue classification with gene expression profiles. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pag. 54–64, New York, NY, USA. ACM Press. (Citado na página 47.)
- Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2:353–373. (Citado na página 25.)
- Braun, H. & Weisbrod, J. (1993). Evolving neural feedforward networks. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pag. 25–32. Springer-Verlag. (Citado nas páginas 20 e 85.)
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32. (Citado na página 82.)
- Carvalho, A. C. P. L. F., Braga, A. P., & Ludermir, T. B. (2003). Computação evolutiva. In Rezende, S. O., editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, capítulo 9, pag. 225–248. Editora Manole Ltda. (Citado na página 32.)
- Carvalho, A. C. P. L. F., Delbem, A. C. B., Romero, R. A. F., Simões, E., & sP. Telles, G. (2004). Computação bioinspirada. Apostila do Minicurso da XXIII Jornada de Atualização em Informática. (Citado nas páginas 23 e 31.)

- Castillo, P. A., Merelo, J. J., Arenas, M. G., & Romero, G. (2007). Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters. *Information Sciences*, 177(14):2884–2905. (Citado nas páginas 20 e 85.)
- Castro, L. N. (2006). *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. Chapman & Hall/CRC. (Citado nas páginas 24, 27, 28 e 30.)
- Castro, L. N. (2007). Fundamentals of natural computing: an overview. *Physics of Life Reviews*, 4(1):1–36. (Citado nas páginas 2 e 23.)
- Castro, L. N. & Timmis, J. (2002). Artificial immune systems: A novel paradigm to pattern recognition. In Corchado, J. M., Alonso, L., & Fyfe, C., editors, *Artificial Neural Networks in Pattern Recognition*, pag. 67–84. University of Paisley. (Citado nas páginas 3, 29 e 30.)
- Castro, L. N. & Von-Zuben, F. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251. (Citado nas páginas 3, 30, 35 e 38.)
- Castrogiovanni, M., Nicosia, G., & Rascunà, R. (2007). Experimental analysis of the aging operator for static and dynamic optimisation problems. In Apolloni, B., Howlett, R. J., & Jain, L. C., editors, *Proceedings of 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science, pag. 804–811. Springer. (Citado na página 38.)
- Chang, C.-C. & Lin, C.-J. (2001). *LIBSVM: a Library for Support Vector Machines*. Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. (Citado na página 19.)
- Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. MIT Press. (Citado na página 7.)
- Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159. (Citado nas páginas 18 e 19.)
- Chunhong, Z. & Licheng, J. (2004). Automatic parameters selection for SVM based on GA. In *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pag. 1869–1872. IEEE Computer Society Press. (Citado na página 2.)
- Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pag. 115–123. (Citado na página 2.)

- Cristianini, N. & Shawe-Taylor, J. (2000). *An introduction to support vector machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA. (Citado nas páginas 3, 14 e 19.)
- Cutello, V., Nicosia, G., & Pavone, M. (2006). Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator. In *Proceedings of the ACM Symposium on Applied Computing*, pag. 950–954, New York, NY, USA. ACM. (Citado na página 38.)
- Daelemans, W., Zavrel, J., van der Sloot, K., & van den Bosch, A. (2002). Timbl: Tilburg memory-based learner, version 4.3, reference guide. Technical Report ILK 02-10, Tilburg University. (Citado na página 2.)
- De-Robertis, E. M. F. & Hib, J. (2001). *Bases da Biologia Celular e Molecular*. Guanabara Koogan, Rio de Janeiro, 3 edition. (Citado na página 46.)
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30. (Citado nas páginas 4, 14, 50, 54, 61, 70, 71, 77 e 78.)
- Diaz-Uriarte, R. & Alvarez de Andres, S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3. (Citado na página 82.)
- Dodd, N. (1990). Optimisation of network structure using genetic techniques. In *Proceedings of the International Joint Conference on Neural Networks*, pag. 965–970. (Citado nas páginas 20 e 85.)
- Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization: Artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, 1(4):28–39. (Citado nas páginas 23 e 25.)
- Dorigo, M. & Di-Caro, G. (1999). The ant colony optimization metaheuristic. In Corne, D., Dorigo, M., & Glover, F., editors, *New Ideas in Optimization*, pag. 11–32. McGraw Hill, London, UK. (Citado na página 25.)
- Draghici, S. & Potter, R. B. (2003). Predicting HIV drug resistance with neural networks. *Bioinformatics*, 19(1):98–107. (Citado na página 16.)
- Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87. (Citado na página 47.)

- Eberhart, R. C. & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pag. 84–88. (Citado na página 38.)
- Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, 19(1):43–53. (Citado nas páginas 35 e 38.)
- Elshamy, W., Emará, H. M., & Bahgat, A. (2007). Clubs-based particle swarm optimization. In *IEEE Swarm Intelligence Symposium*, pag. 289–296. (Citado nas páginas 36, 40, 41 e 42.)
- Eshelman, L. J. & Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. In Whitley, L. D., editor, *Foundation of Genetic Algorithms 2*, pag. 187–202. Morgan Kaufmann Publishers, San Mateo. (Citado na página 33.)
- Eusuff, M. M. & Lansey, K. E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 129(3):210–225. (Citado na página 35.)
- Faceli, K., Carvalho, A. C. P. L. F., & Souto, M. C. P. (2005). Análise de dados de expressão gênica. Technical report, Universidade de São Paulo - ICMC, São Carlos. Relatório Técnico 250. (Citado nas páginas 46 e 47.)
- Fahlman, S. E. (1988). Faster-learning variations on back-propagation: An empirical study. In *Proceedings of the 1988 Connectionist Models Summer School*, Los Altos, CA. Morgan Kaufmann. (Citado na página 20.)
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874. ROC Analysis in Pattern Recognition. (Citado na página 12.)
- Freitas, A. A. (2003). A survey of evolutionary algorithms for data mining and knowledge discovery. In Ghosh, A. & Tsutsui, S., editors, *Advances in evolutionary computing: theory and applications*, pag. 819–845. Springer-Verlag New York, Inc., New York, NY, USA. (Citado na página 35.)
- Gao, L., Zhou, C., Gao, H.-B., & Shi, Y.-R. (2006). Credit scoring model based on neural network with particle swarm optimization. In *Proceedings of the Second International Conference on Advances in Natural Computation*, pag. 76–79. Springer-Verlag. (Citado nas páginas 20 e 85.)
- Glover, F. & Laguna, F. (1997). *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA. (Citado nas páginas 19 e 24.)

- Goldberg, D. E. & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3(2-3):95–99. (Citado nas páginas 3 e 31.)
- Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural Network Design*. PWS Publishing, Boston, 1 edition. (Citado na página 17.)
- Haslinger, C., Schweifer, N., Stilgenbauer, S., Dohner, H., Lichter, P., Kraut, N., Stratowa, C., & Abseher, R. (2004). Microarray Gene Expression Profiling of B-Cell Chronic Lymphocytic Leukemia Subgroups Defined by Genomic Aberrations and VH Mutation Status. *Journal of Clinical Oncology*, 22(19):3937–3949. (Citado na página 46.)
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. (Citado nas páginas xxi, 3, 14, 15 e 16.)
- Herrera, F., Herrera-Viedma, E., Lozano, M., & Verdegay, J. L. (1994). Fuzzy tools to improve genetic algorithms. In *Proceedings of the Second European Congress on Intelligent Techniques and Soft Computing*, pag. 1532–1539. (Citado na página 33.)
- Herrera, F., Lozano, M., & Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, 12(4):265–319. (Citado nas páginas 32, 33 e 34.)
- Hintz, K. J. & Spofford, J. J. (1990). Evolving a neural network. In *Proceedings of the IEEE International Symposium on Intelligent Control*, pag. 479–484. (Citado na página 20.)
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor. (Citado nas páginas 24 e 31.)
- Hoste, V. & Daelemans, W. (2005). Comparing learning approaches to coreference resolution. There is more to it than 'bias'. In *Proceedings of the Workshop on Meta-Learning*, pag. 20–27. (Citado nas páginas 2, 3 e 85.)
- Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2007). *A Practical Guide to Support Vector Classification*. Department of Computer Science - National Taiwan University, Taipei, Taiwan. (Citado nas páginas 49, 52, 64 e 90.)
- Huang, C.-L. & Wang, C.-J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, 31(2):231–240. (Citado nas páginas 19, 38 e 85.)
- Hutter, F. & Hamadi, Y. (2005). Parameter adjustment based on performance prediction: Towards an instance-aware problem solver. Technical report, Microsoft Research, Redmond, WA. (Citado na página 18.)

- Ide, A. & Yasuda, K. (2005). A basic study of adaptive particle swarm optimization. *Electrical Engineering in Japan*, 151(3):41–49. (Citado nas páginas 42 e 90.)
- Imbault, F. & Lebart, K. (2004). A stochastic optimization approach for parameter tuning of support vector machines. In *Proceedings of the 17th International Conference on Pattern Recognition*, pag. 597–600. (Citado nas páginas 19 e 85.)
- Ishikawa, M., Yoshida, K., Yamashita, Y., Ota, J., Takada, S., Kisanuki, H., Koinuma, K., Choi, Y. L., Kaneda, R., Iwao, T., Tamada, K., Sugano, K., & Mano, H. (2005). Experimental trial for diagnosis of pancreatic ductal carcinoma based on gene expression profiles of pancreatic ductal cells. *Cancer Science*, 96(7):387–393. (Citado na página 46.)
- Kearns, M., Mansour, Y., Ng, A. Y., & Ron, D. (1997). An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27:7–50. (Citado nas páginas 17 e 18.)
- Kennedy, J. (2005). Particle swarms: optimization based on sociocognition. In Castro, L. & Zuben, F. V., editors, *Recent Development in Biologically Inspired Computing*, pag. 235–269. Idea Group. (Citado na página 35.)
- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pag. 1942–1948, Perth, Australia. (Citado nas páginas 3, 26 e 38.)
- Kennedy, J. & Eberhart, R. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers. (Citado na página 27.)
- Kim, D. H. (2002). Parameter tuning of fuzzy neural networks by immune algorithm. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pag. 408–413. (Citado na página 21.)
- Kirkpatrick, S., Gelatt, C. D., J., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598):671–680. (Citado na página 19.)
- Kohavi, R. & John, G. H. (1995). Automatic parameter selection by minimizing estimated error. In Prieditis, A. & Russel, S., editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pag. 304–312, San Francisco, CA. Morgan Kaufmann. (Citado nas páginas 1 e 18.)
- Lacerda, E. G. M., Carvalho, A. C. P. L. F., & Ludermir, T. B. (2002). Model selection via genetic algorithms for RBF networks. *Journal of Intelligent and Fuzzy Systems*, 13(2-4):111–122. (Citado na página 2.)

- Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armananzas, R., Santafe, G., Perez, A., & Robles, V. (2006). Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112. (Citado na página 10.)
- Leung, F. H. F., Lam, H. K., Ling, S. H., & Tam, P. K. S. (2003). Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks*, 14(1):79–88. (Citado nas páginas 2, 20 e 85.)
- Lorena, A. C. & Carvalho, A. C. P. L. F. (2006). Multiclass SVM design and parameter selection with genetic algorithms. In *Proceedings of the Ninth Brazilian Symposium on Neural Networks*, pag. 23, Washington, DC, USA. IEEE Computer Society. (Citado nas páginas 2, 19 e 85.)
- Lourenço, H. R., Martin, O., & Stutzle, T. (2002). Iterated local search. In Glover, F. & Kochenberger, G., editors, *Handbook of Metaheuristics*, volume 57, pag. 321–353. Kluwer Academic Publishers. (Citado nas páginas 24, 42 e 90.)
- Maniezzo, V., Gambardella, L. M., & Luigi, F. (2004). Ant colony optimization. In Onwubolu, G. C. & Babu, B. V., editors, *New Optimization Techniques in Engineering*, pag. 101–117. Springer-Verlag, Berlin, Heidelberg. (Citado nas páginas 24 e 26.)
- McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133. (Citado nas páginas 16 e 17.)
- Meyer, D. (2001). Support vector machines. *R News*, 1(3):23–26. (Citado na página 49.)
- Mezura-Montes, E. & Lopez-Ramirez, B. C. (2007). Comparing bio-inspired algorithms in constrained optimization problems. In *IEEE Congress on Evolutionary Computation*, pag. 662–669. (Citado na página 35.)
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag. (Citado na página 33.)
- Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Introduction. In Michie, D., Spiegelhalter, D. J., & Taylor, C. C., editors, *Machine Learning, Neural and Statistical Classification*. Ellis Horwood. (Citado na página 47.)
- Miller, G. F., Todd, P. M., & Hegde, S. U. (1989). Designing neural networks using genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pag. 379–384, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Citado na página 20.)
- Millonas, M. M. (1994). Swarms, phase transitions, and collective intelligence. In Langton, C. G., editor, *Artificial Life III*, volume XVII of *Santa Fe Institute Studies in the*

- Sciences of Complexity*, pag. 417–445. Addison-Wesley Longman Publishing, New York, NY. (Citado na página 24.)
- Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18:203–266. (Citado na página 18.)
- Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill, New York. (Citado nas páginas 1, 7, 13 e 15.)
- Monard, M. C. & Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. In Rezende, S. O., editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, capítulo 4, pag. 89–114. Editora Manole Ltda. (Citado nas páginas 1, 7, 9 e 10.)
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report Caltech Concurrent Computation Program 826, California Institute of Technology, Pasadena, California, USA. (Citado na página 35.)
- Nadeau, C. & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, 52(3):239–281. (Citado nas páginas 14, 50 e 54.)
- Narayanan, A., Keedwell, E. C., & Olsson, B. (2002). Artificial intelligence techniques for bioinformatics. *Applied Bioinformatics*, 1(4):191–222. (Citado na página 16.)
- Nolfi, S. & Floreano, D. (2004). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Bradford Book. (Citado na página 35.)
- Nutt, C. L., Mani, D. R., Betensky, R. A., Tamayo, P., Cairncross, J. G., Ladd, C., Pohl, U., Hartmann, C., McLaughlin, M. E., Batchelor, T. T., Black, P. M., von Deimling, A., Pomeroy, S. L., Golub, T. R., & Louis, D. N. (2003). Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Research*, 63(7):1602–1607. (Citado na página 46.)
- Perelson, A. S. & Oster, G. (1979). Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-nonsel discrimination. *Journal of Theoretical Biology*, 81(4):645–670. (Citado na página 30.)
- Phadke, M. S. (1995). *Quality Engineering Using Robust Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA. (Citado na página 20.)
- Price, K., Storn, R. M., & Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. (Citado na página 35.)

- Rechenberg, I. (1973). *Evolutionstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog. (Citado na página 35.)
- Rifkin, R., Mukherjee, S., Tamayo, P., Ramaswamy, S., Yeang, C.-H., Angelo, M., Reich, M., Poggio, T., Lander, E. S., Golub, T. R., & Mesirov, J. P. (2003). An analytical method for multi-class cancer classification. *SIAM Review*, 45(4):706–723. (Citado na página 82.)
- Rosenbrock, H. H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184. (Citado na página 36.)
- Rossi, A. L. D. & Carvalho, A. C. P. L. F. (2008). Bio-inspired optimization techniques for SVM parameter tuning. In *Proceedings of 10th Brazilian Symposium on Neural Networks*, pag. 435–440. IEEE Computer Society. (Citado na página 89.)
- Rossi, A. L. D., Carvalho, A. C. P. L. F., & Soares, C. (2008a). Bio-inspired parameter tuning of MLP networks for gene expression analysis. In *Proceedings of 8th International Conference on Hybrid Intelligent Systems*, pag. 57–62. IEEE Computer Society. (Citado na página 89.)
- Rossi, A. L. D., Soares, C., & Carvalho, A. C. P. L. F. (2008b). Bioinspired parameter tuning of MLP networks for gene expression analysis: quality of fitness estimates vs number of solutions analyzed. In *Proceedings of 15th International Conference on Neuro- Information Processing of the Asia Pacific Neural Network Assembly*. Aceito. (Citado na página 89.)
- Schaffer, J. D., Caruana, R. A., Eshelman, L. J., & Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In Schaffer, J., editor, *Proceedings of the Third international Conference on Genetic Algorithms*, pag. 51–60, San Francisco, CA. Morgan Kaufmann Publishers. (Citado na página 36.)
- Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270:467–470. (Citado na página 47.)
- Schwarz, G. (1978). Estimating the dimension of a model. In *Annals of Statistics*, volume 6, pag. 461–464. (Citado na página 18.)
- Seiffert, U., Hammer, B., Kaski, S., & Villmann, T. (2006). Neural networks and machine learning in bioinformatics - theory and applications. In *Proceedings of the European Symposium on Artificial Neural Networks*, pag. 521–532, Bruges, Belgium. (Citado na página 16.)

- Shi, Y. & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pag. 69–73, Anchorage, Alaska. (Citado nas páginas 29, 36 e 38.)
- Socha, K. (2004). ACO for continuous and mixed-variable optimization. In Dorigo, M., Birattari, M., Blum, C., Gambardella, L. M., Mondada, F., & Stützle, T., editors, *Proceedings of the 4th International Workshop on Ant Colony Optimization*, volume 3172 of *Lecture Notes in Computer Science*, pag. 25–36. Springer. (Citado nas páginas 26, 27 e 38.)
- Socha, K. & Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3):1155–1173. (Citado nas páginas 3, 26, 27, 35, 38 e 48.)
- Sousa, T., Silva, A., & Neves, A. (2004). Particle swarm based data mining algorithms for classification tasks. *Parallel Comput.*, 30(5-6):767–783. (Citado na página 35.)
- Souto, M. C. P., Lorena, A. C., Delbem, A. C. B., & Carvalho, A. C. P. L. F. (2003). Técnicas de aprendizado de máquina para problemas de biologia molecular. In *II Jornada de Atualização em Inteligência Artificial*, pag. 1–40, Campinas - Brasil. (Citado nas páginas xix, 14, 15, 32 e 33.)
- Souza, B. F. (2005). Seleção de características em SVMs aplicadas a dados de expressão gênica. Master's thesis, Instituto de Ciências Matemáticas e de Computação- Universidade de São Paulo, São Carlos, SP. (Citado na página 47.)
- Souza, B. F. & Carvalho, A. C. P. L. F. (2005). Gene selection based on multi-class support vector machines and genetic algorithms. *Genetics and Molecular Research*, 4(3):599–607. (Citado nas páginas 19 e 85.)
- Souza, B. F., Carvalho, A. C. P. L. F., Calvo, R., & Ishii, R. P. (2006). Multiclass SVM model selection using particle swarm optimization. In *Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*, pag. 31, Washington, DC, USA. IEEE Computer Society. (Citado nas páginas 2, 19 e 85.)
- Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., & Levy, S. (2005a). A comprehensive evaluation of multiclass classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–643. (Citado nas páginas 48, 64, 82 e 90.)
- Statnikov, A., Tsamardinos, I., Dosbayev, Y., & Aliferis, C. F. (2005b). GEMS: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *International Journal of Medical Informatics*, 74(7-8):491 – 503. MedInfo 2004. (Citado na página 64.)

- Statnikov, A., Wang, L., & Aliferis, C. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9(1):319. (Citado nas páginas 81, 82 e 90.)
- Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. Addison-Wesley, Boston, MA, USA. (Citado na página 8.)
- Teräsvirta, T., Lin, C., & Granger, C. W. J. (1993). Power of the neural network linearity test. *Journal of Time Series Analysis*, 14(2):209–220. (Citado na página 18.)
- Trelea, I. C. (2003). The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325. (Citado na página 62.)
- Tsai, J.-T., Chou, J.-H., & Liu, T.-K. (2006). Tuning the structure and parameters of a neural network by using hybrid taguchi-genetic algorithm. *IEEE Transactions on Neural Networks*, 17(1):69–80. (Citado nas páginas 20 e 85.)
- Vapnik, V. & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of occurrence of events to their probabilities. *Theory of Probability and Its Applications*, 2(16):264–280. (Citado na página 14.)
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag. (Citado na página 14.)
- Velculescu, V. E., Zhang, L., Vogelstein, B., & Kinzler, K. W. (1995). Serial analysis of gene expression. *Science*, 270:484–487. (Citado na página 47.)
- Wang, J. T. L., Ma, Q., Shasha, D., & Wu, C. H. (2000). Application of neural networks to biological data mining: A case study in protein sequence classification. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pag. 305–309, New York, NY, USA. ACM Press. (Citado na página 16.)
- Xu, P. & Chan, A. K. (2003). An efficient algorithm on multi-class support vector machine model selection. In *Proceedings of the International Conference on Neural Networks*, pag. 3229–3232. (Citado na página 18.)
- Yeoh, E.-J., Ross, M. E., Shurtleff, S. A., Williams, W. K., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A., Cheng, C., Campana, D., Wilkins, D., Zhou, X., Li, J., Liu, H., Pui, C.-H., Evans, W. E., Naeve, C., Wong, L., & Downing, J. R. (2002). Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133–143. (Citado na página 46.)

- Yildiz, O. T. & Alpaydin, E. (2005). Model selection in omnivariate decision trees. In *Proceedings of the 16th European Conference on Machine Learning*, volume 3720 of *Lecture Notes in Computer Science*, pag. 473–484. Springer. (Citado na página 18.)
- Zhang, X. & Jiao, L. (2005). Simultaneous feature selection and parameters optimization for SVM by immune clonal algorithm. In Wang, L., Chen, K., & Ong, Y.-S., editors, *ICNC (2)*, volume 3611 of *Lecture Notes in Computer Science*, pag. 905–912. Springer. (Citado na página 85.)

Apêndice A

Densidade das Taxas de Erro de Validação e de Teste

Neste apêndice são apresentados os gráficos referentes às densidades das taxas de erro de validação e de teste para todas as combinações de parâmetros utilizados nos experimentos da Seção 5.7. Os gráficos exibidos nas figuras de A.1 a A.4 mostram as densidades para os conjuntos de dados utilizados nos experimentos com RNs, descritos na Seção 5.7.1. Os gráficos exibidos nas figuras de A.5 a A.14 mostram as densidades para os conjuntos de dados utilizados nos experimentos com SVMs, reportados na Seção 5.7.2.

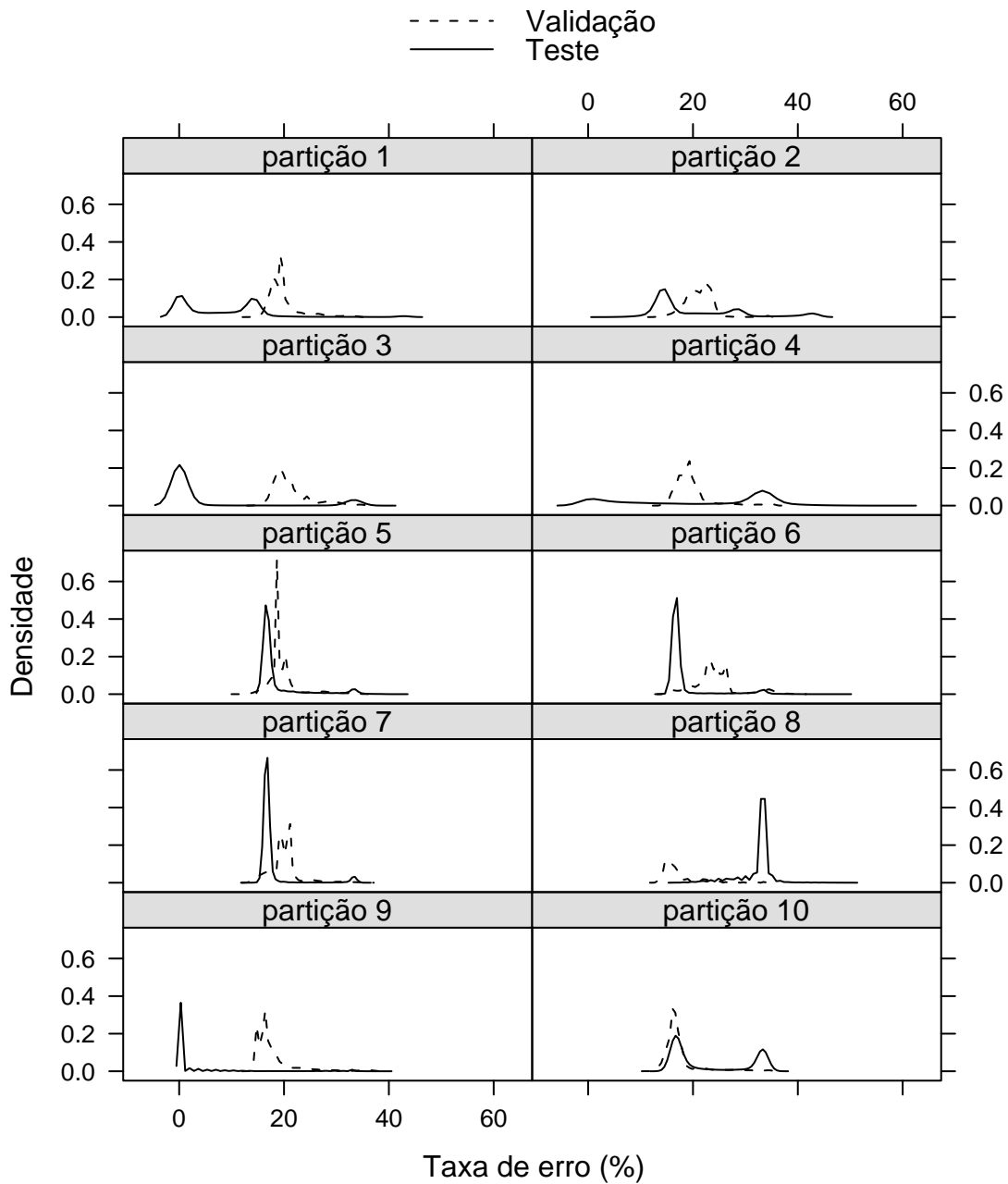


Figura A.1: Densidade da taxa de erro de validação e de teste para o conjunto de dados Cólón para todas as combinações de valores de parâmetros para RNs.

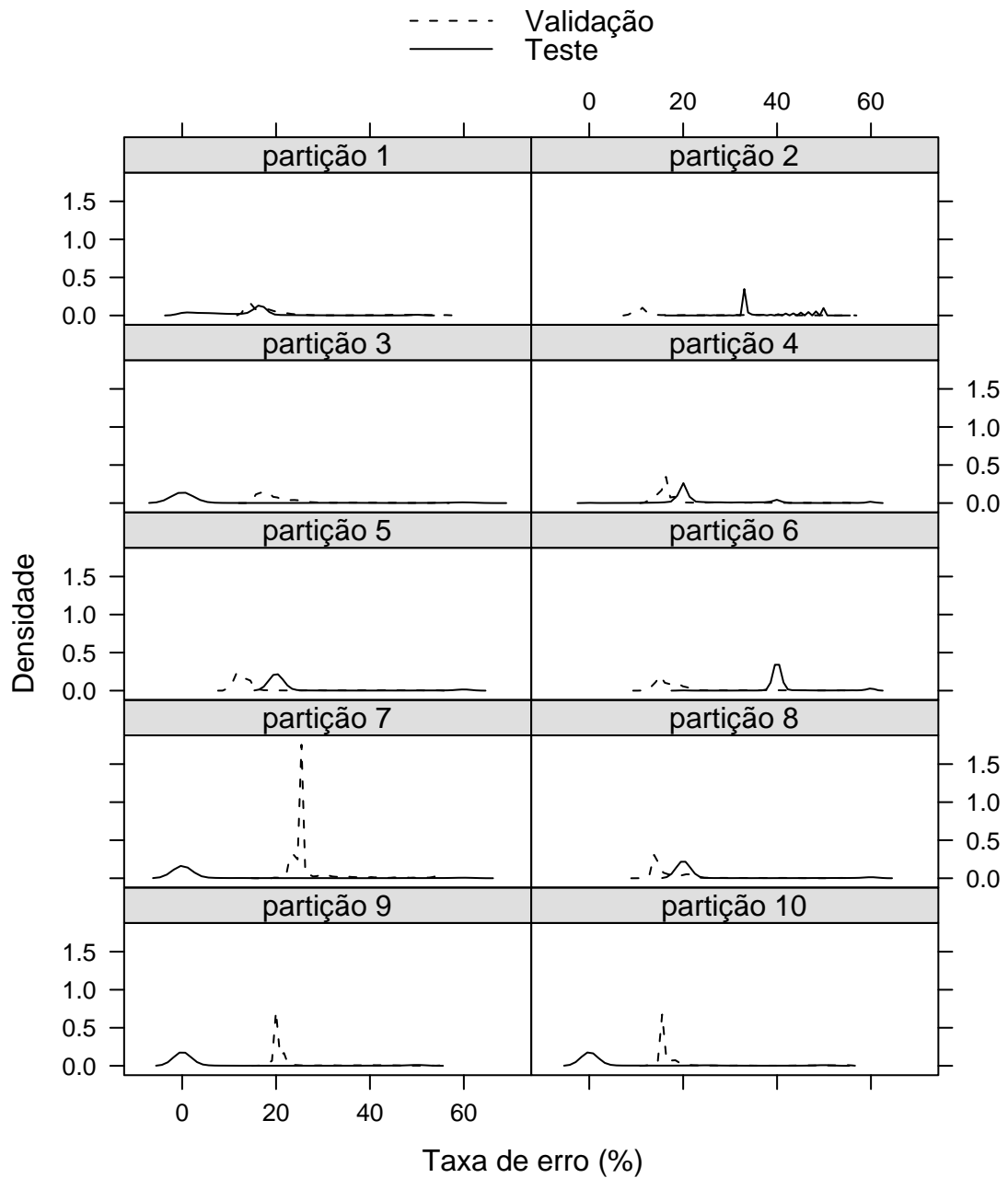


Figura A.2: Densidade da taxa de erro de validação e de teste para o conjunto de dados Glioma para todas as combinações de valores de parâmetros para RNs.

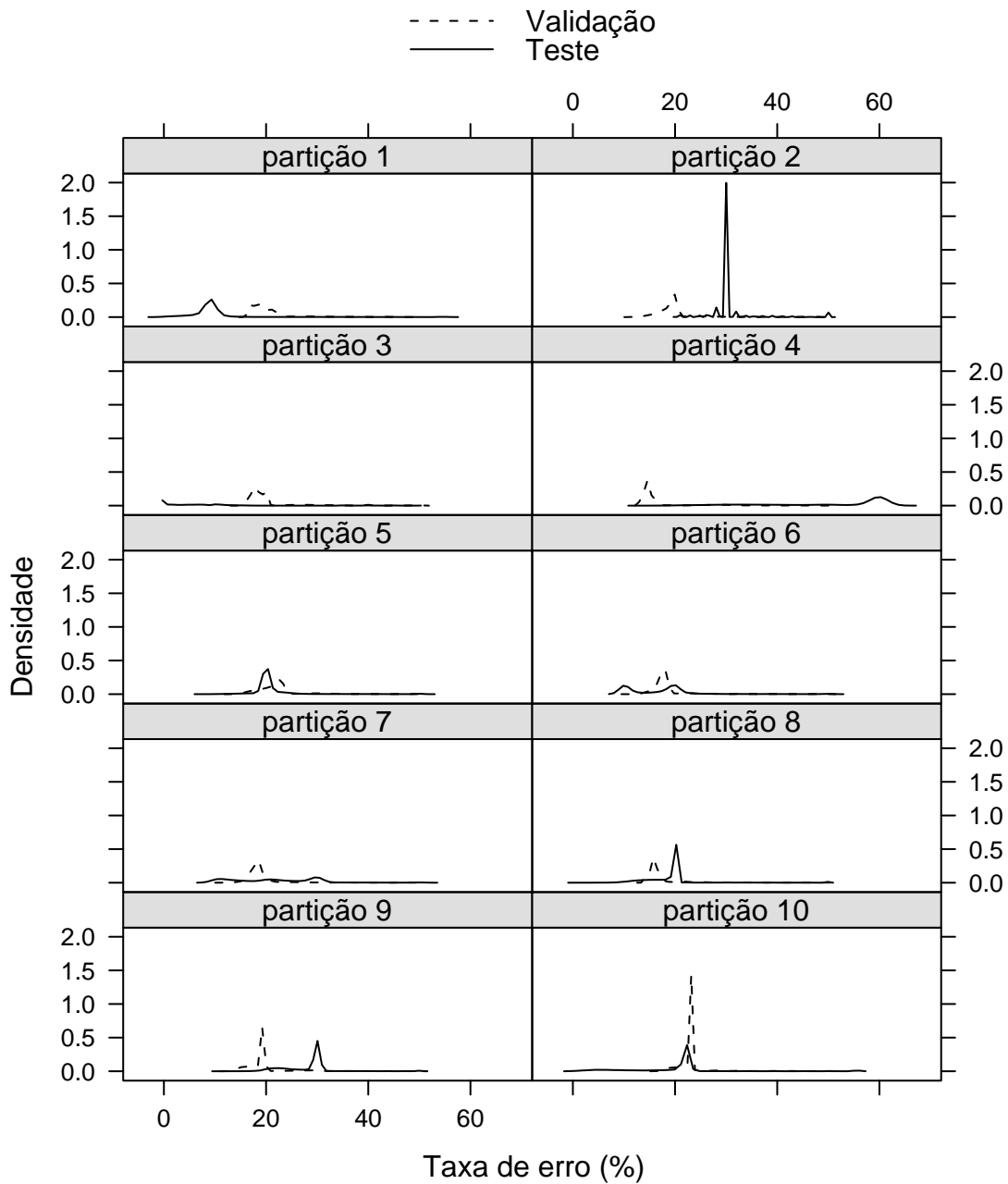


Figura A.3: Densidade da taxa de erro de validação e de teste para o conjunto de dados Leucemia para todas as combinações de valores de parâmetros para RNs.

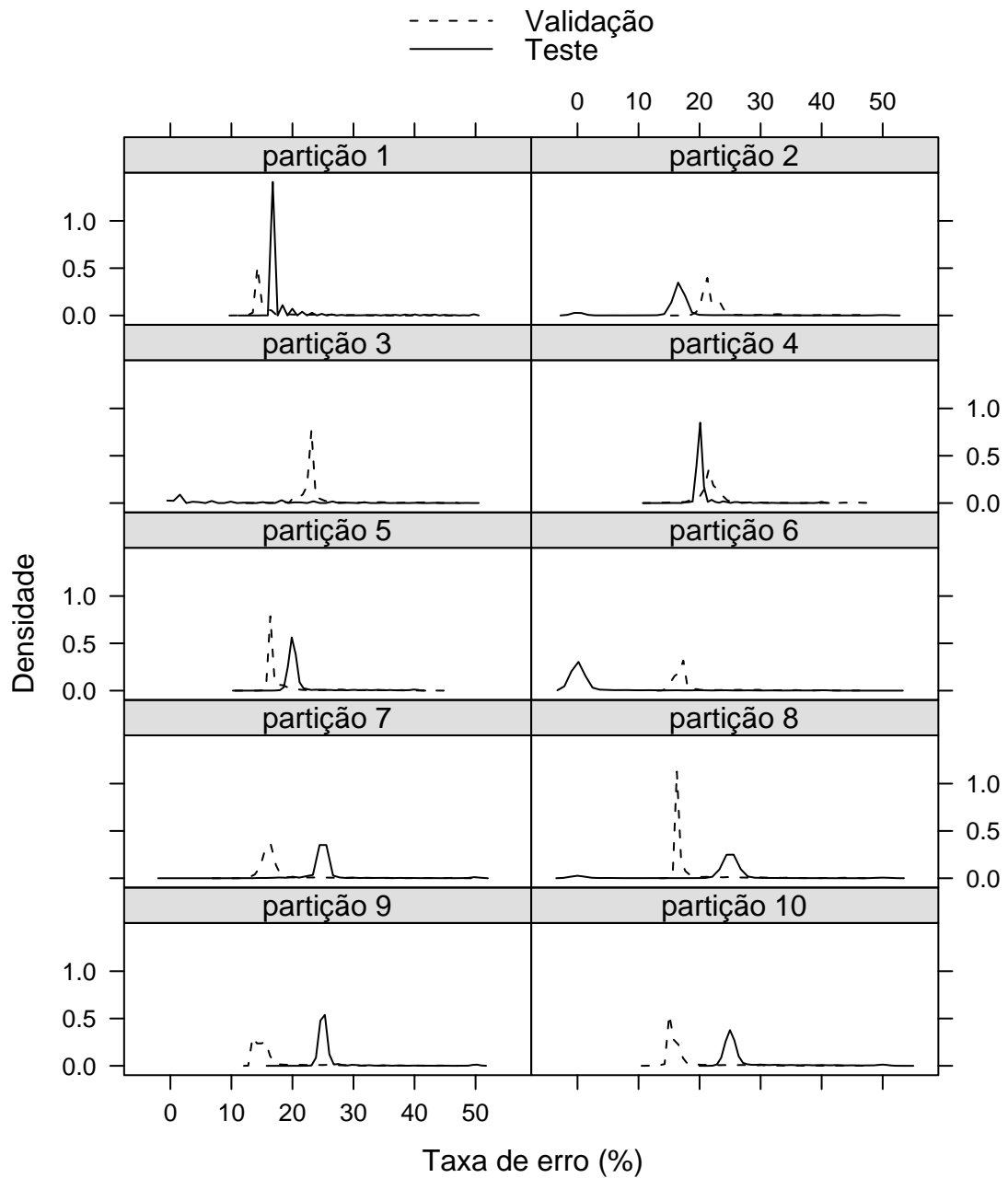


Figura A.4: Densidade da taxa de erro de validação e de teste para o conjunto de dados Pâncreas para todas as combinações de valores de parâmetros para RNs.

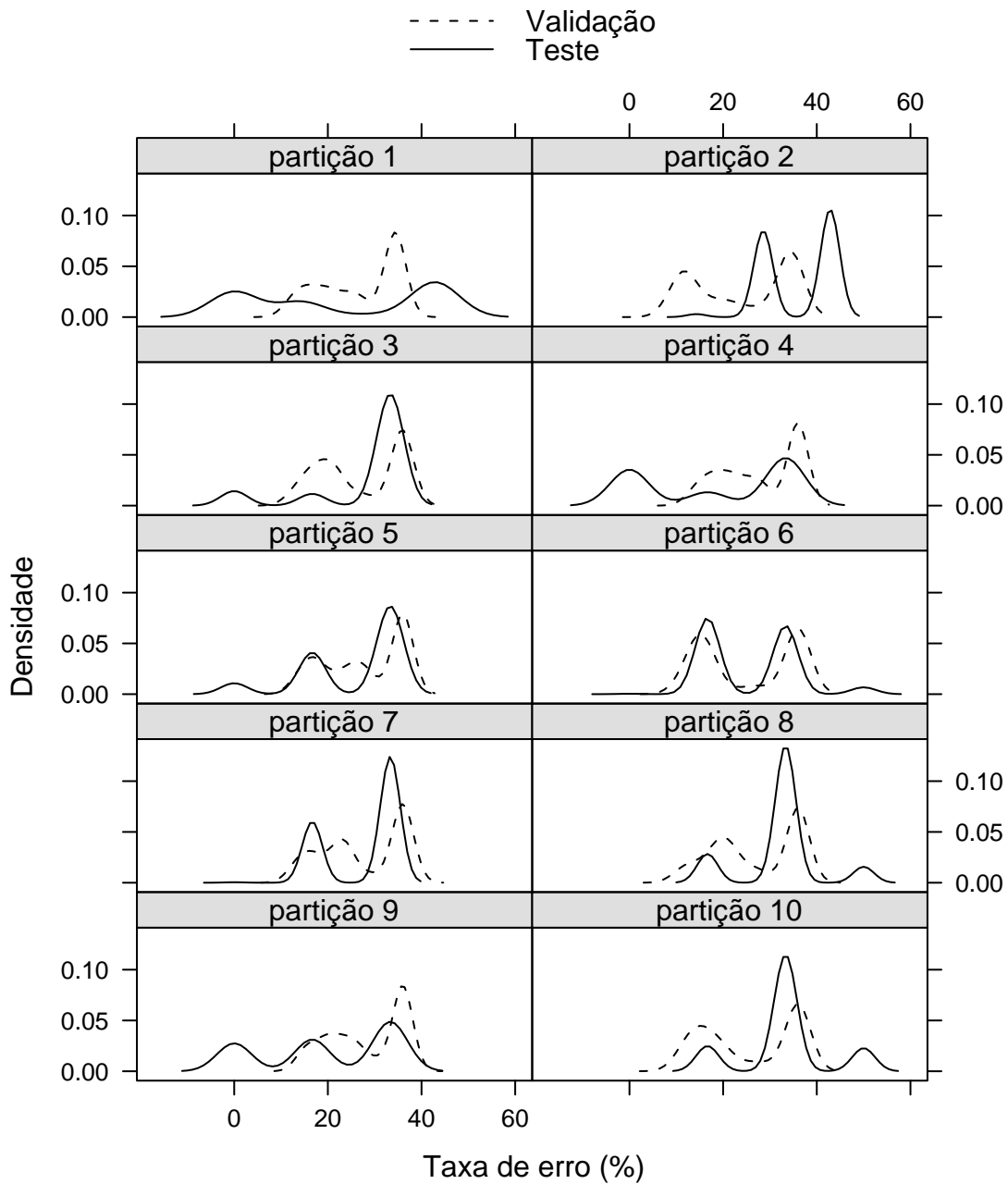


Figura A.5: Densidade da taxa de erro de validação e de teste para o conjunto de dados Cólón para todas as combinações de valores de parâmetros para SVMs.

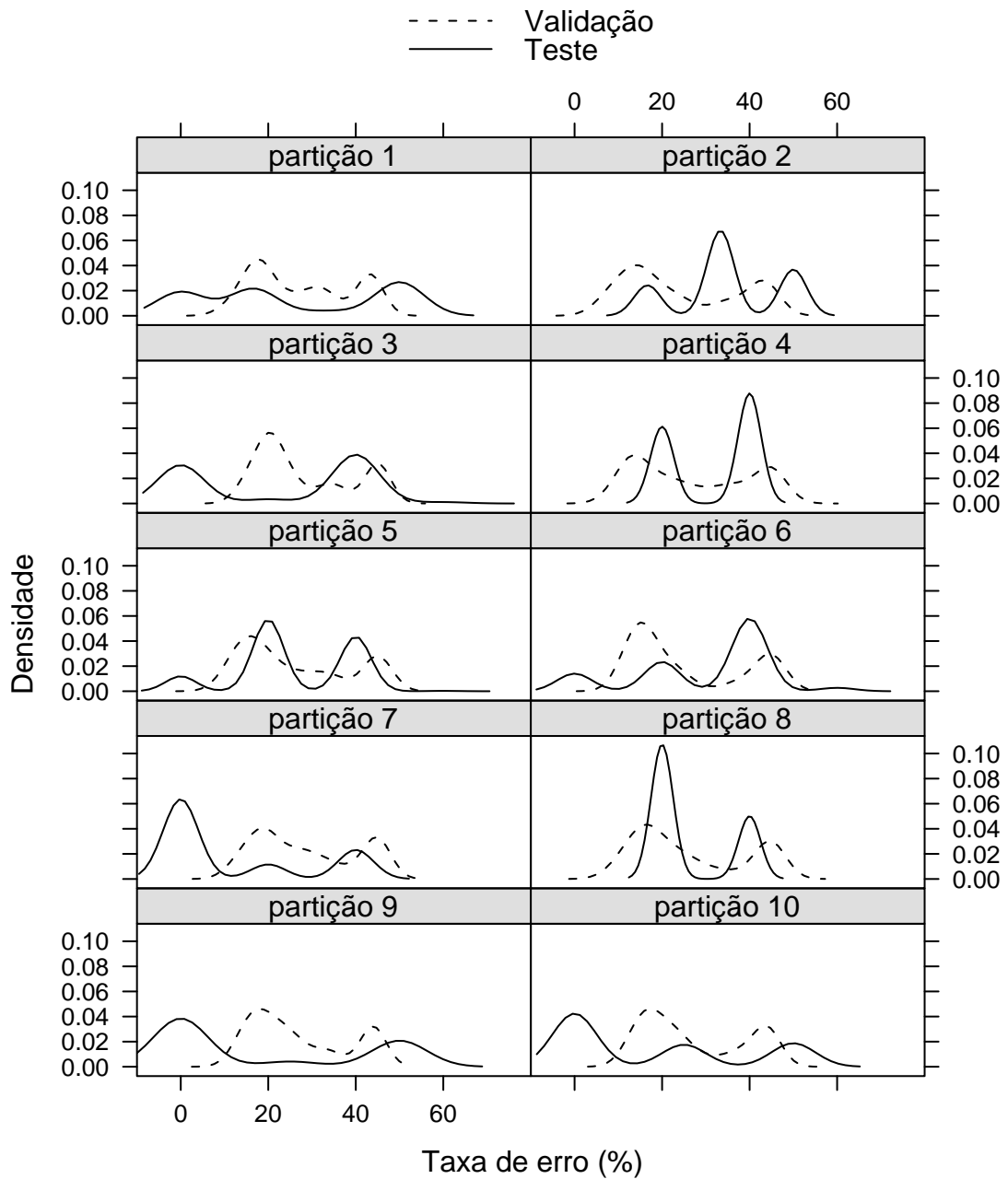


Figura A.6: Densidade da taxa de erro de validação e de teste para o conjunto de dados Glioma para todas as combinações de valores de parâmetros para SVMs.

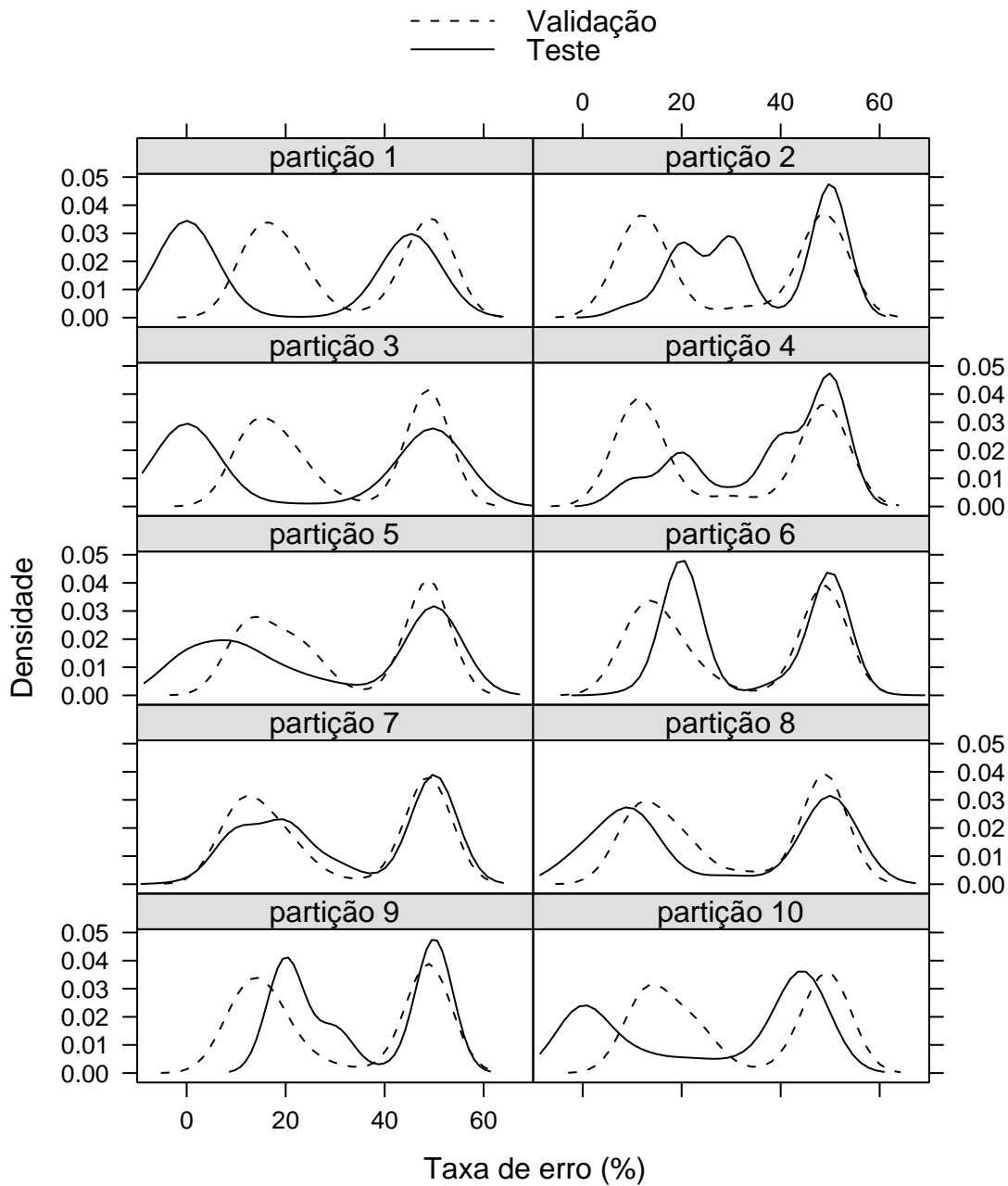


Figura A.7: Densidade da taxa de erro de validação e de teste para o conjunto de dados Leucemia para todas as combinações de valores de parâmetros para SVMs.

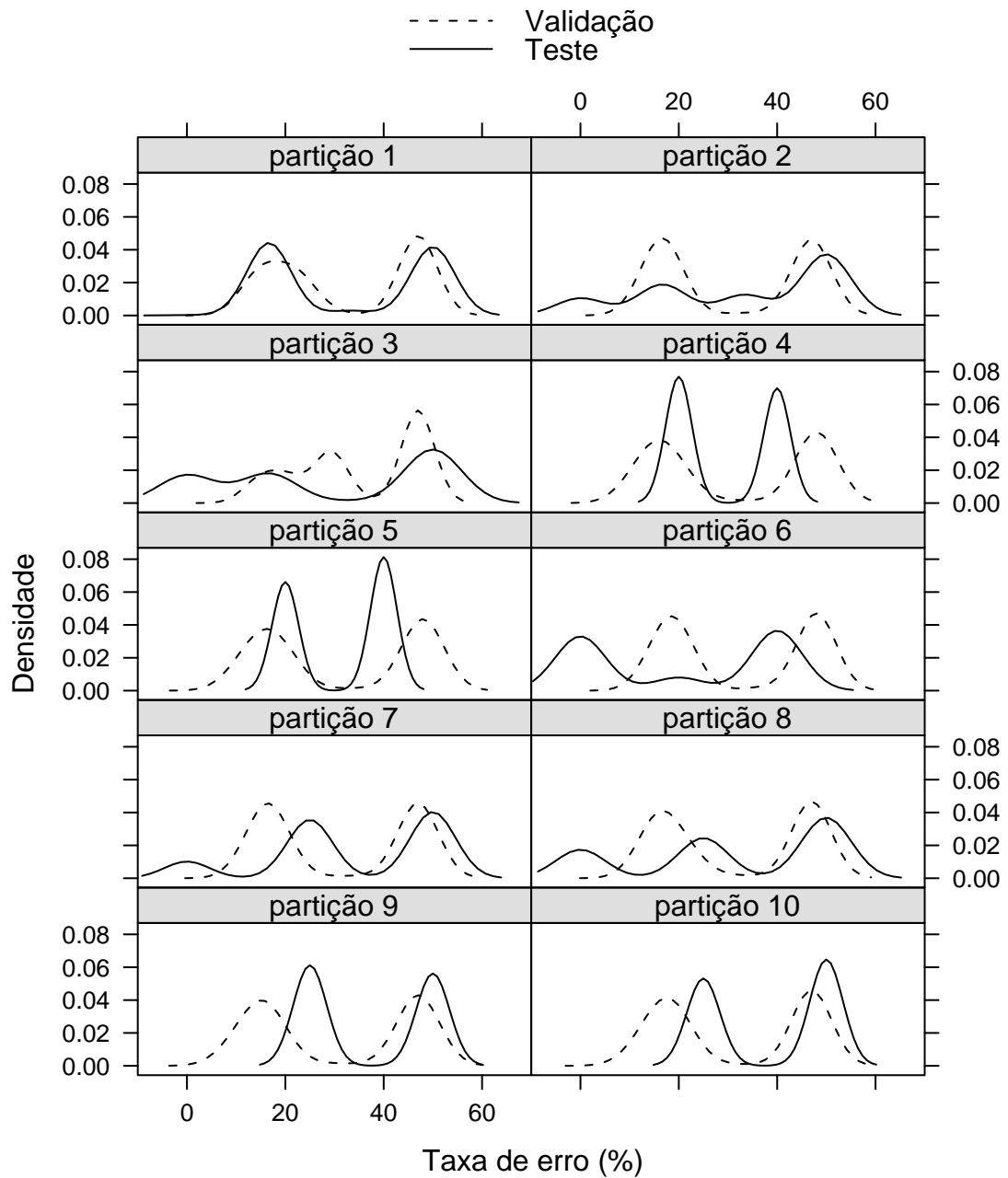


Figura A.8: Densidade da taxa de erro de validação e de teste para o conjunto de dados Pâncreas para todas as combinações de valores de parâmetros para SVMs.

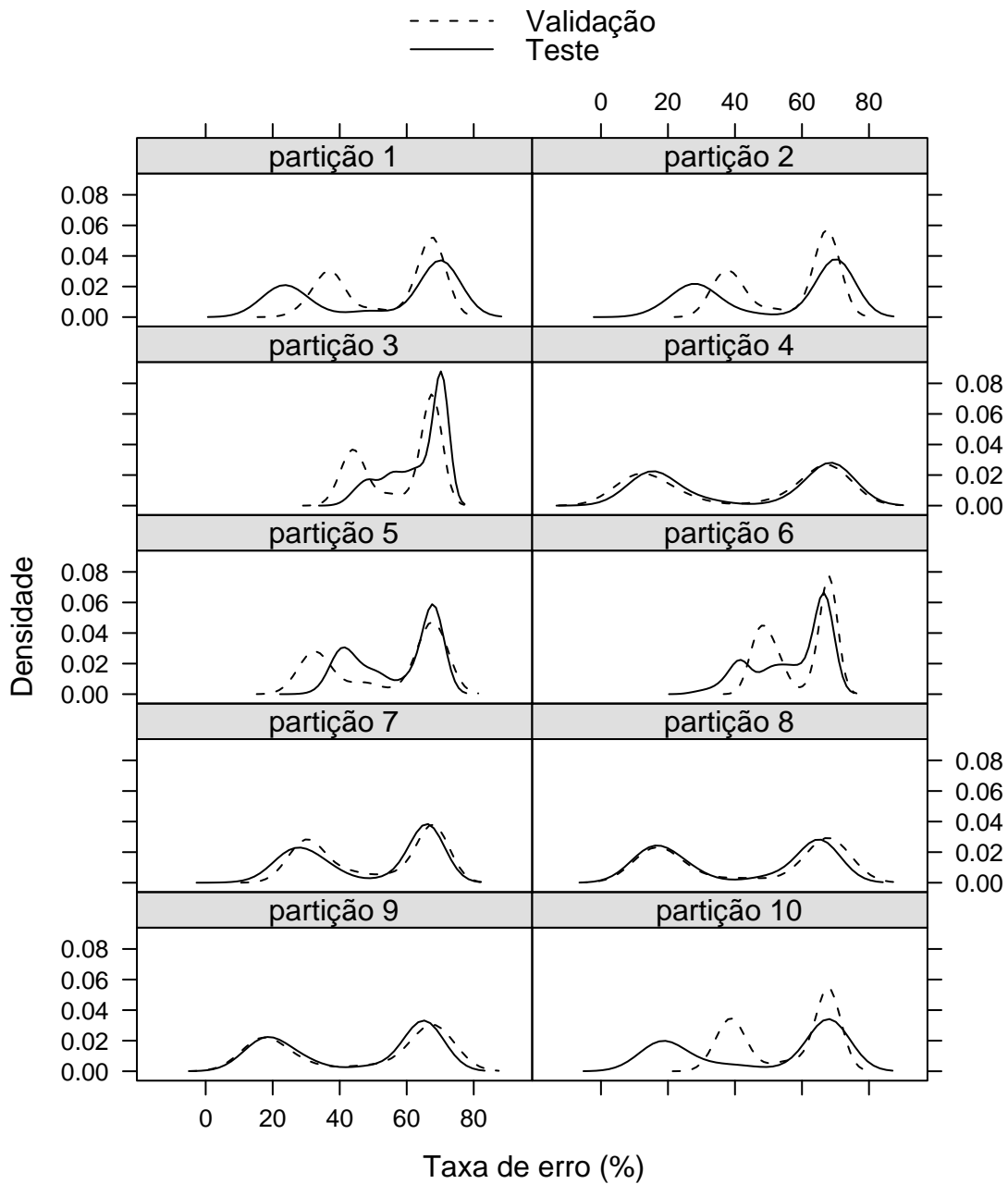


Figura A.9: Densidade da taxa de erro de validação e de teste para o conjunto de dados Leucemia2 para todas as combinações de valores de parâmetros para SVMs.

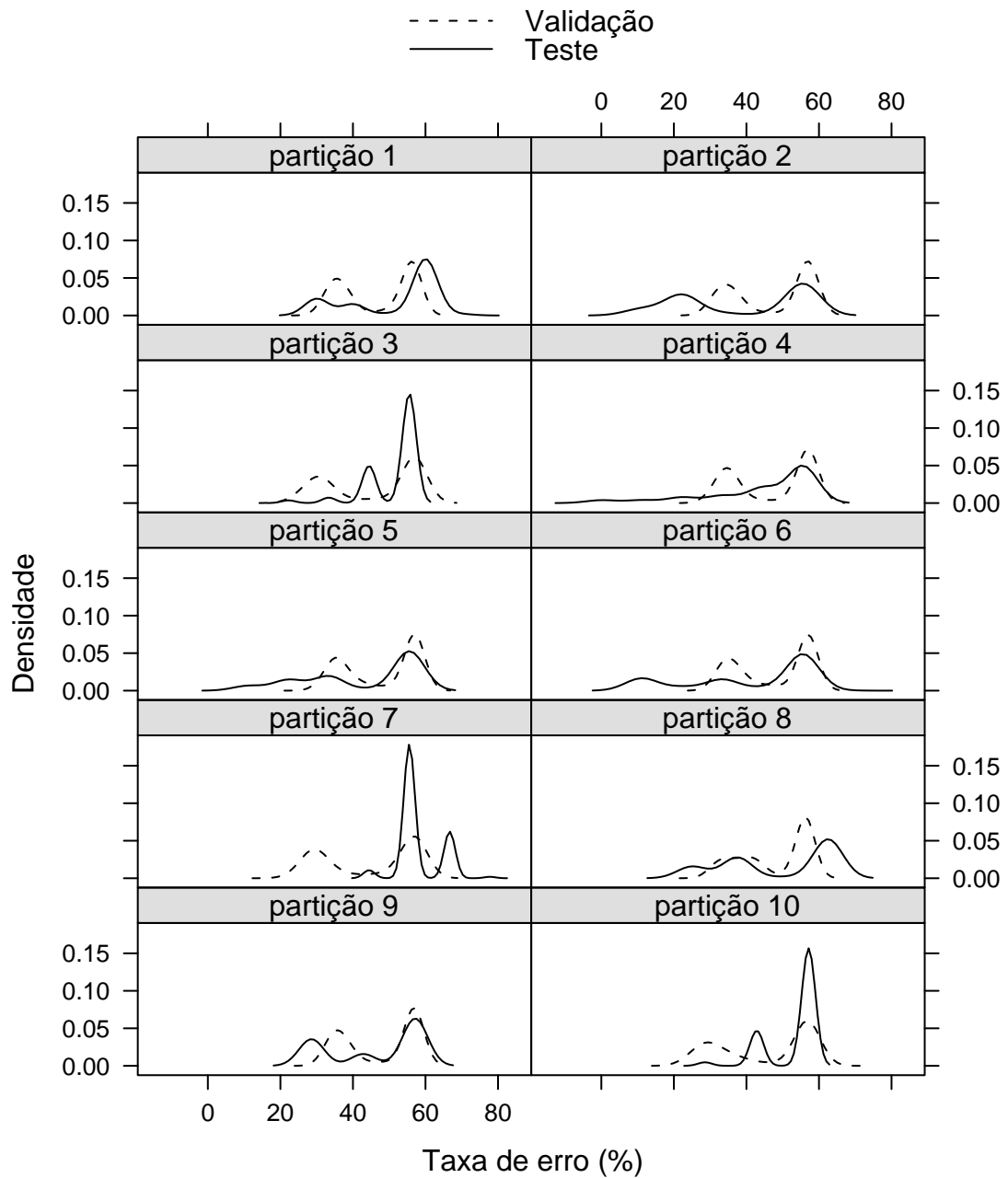


Figura A.10: Densidade da taxa de erro de validação e de teste para o conjunto de dados Pulmão para todas as combinações de valores de parâmetros para SVMs.

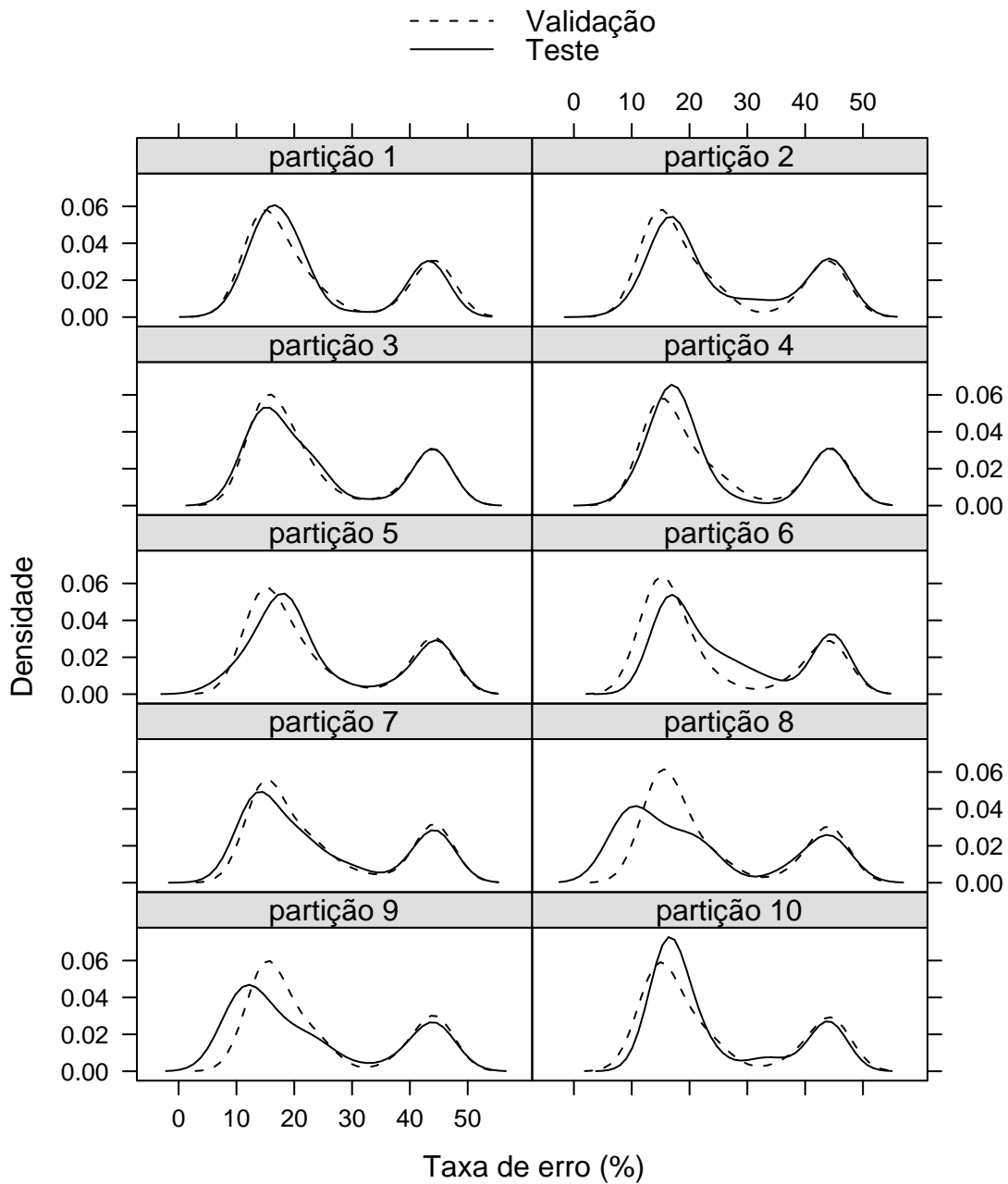


Figura A.11: Densidade da taxa de erro de validação e de teste para o conjunto de dados *australian* para todas as combinações de valores de parâmetros para SVMs.

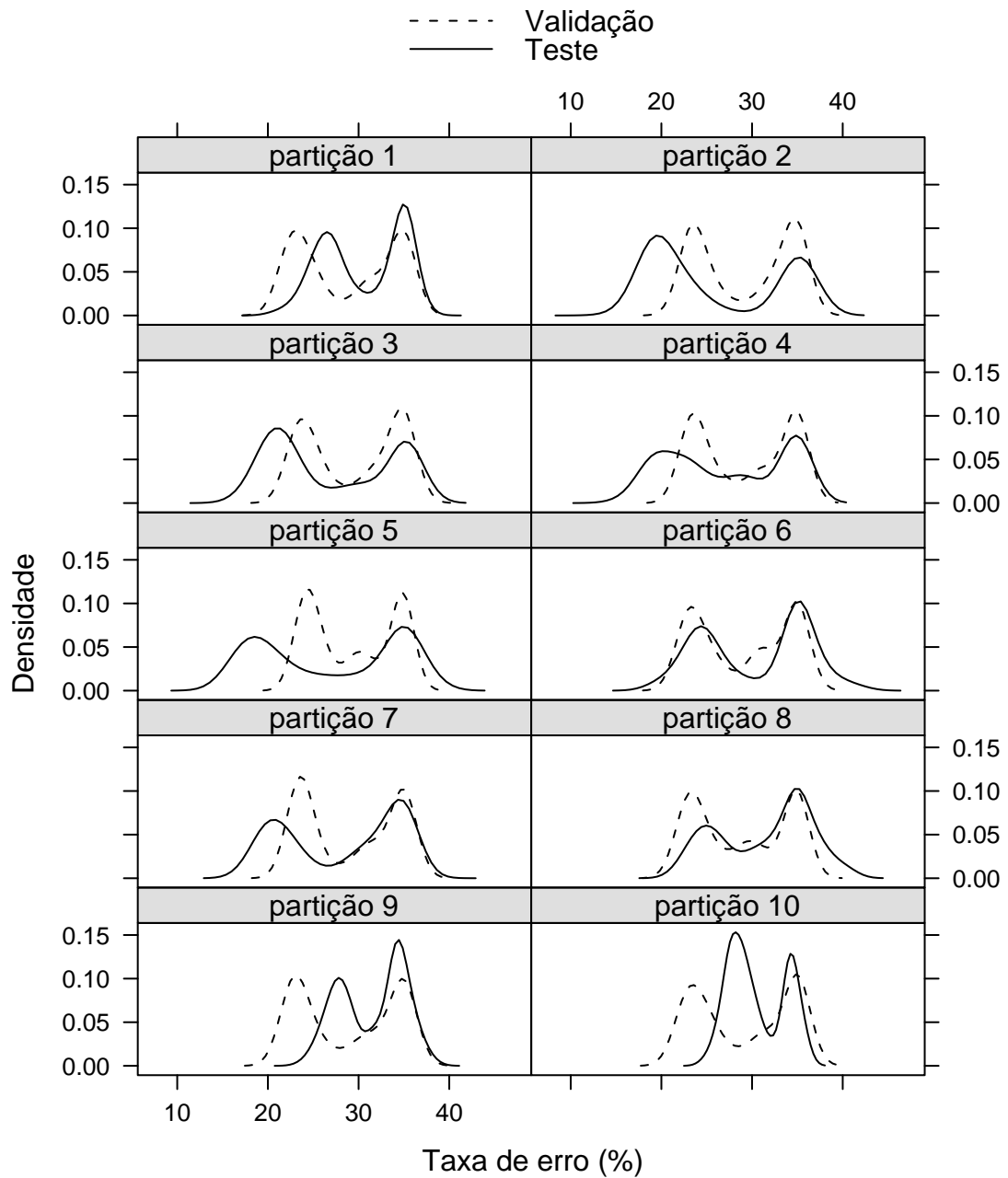


Figura A.12: Densidade da taxa de erro de validação e de teste para o conjunto de dados *pima* para todas as combinações de valores de parâmetros para SVMs.

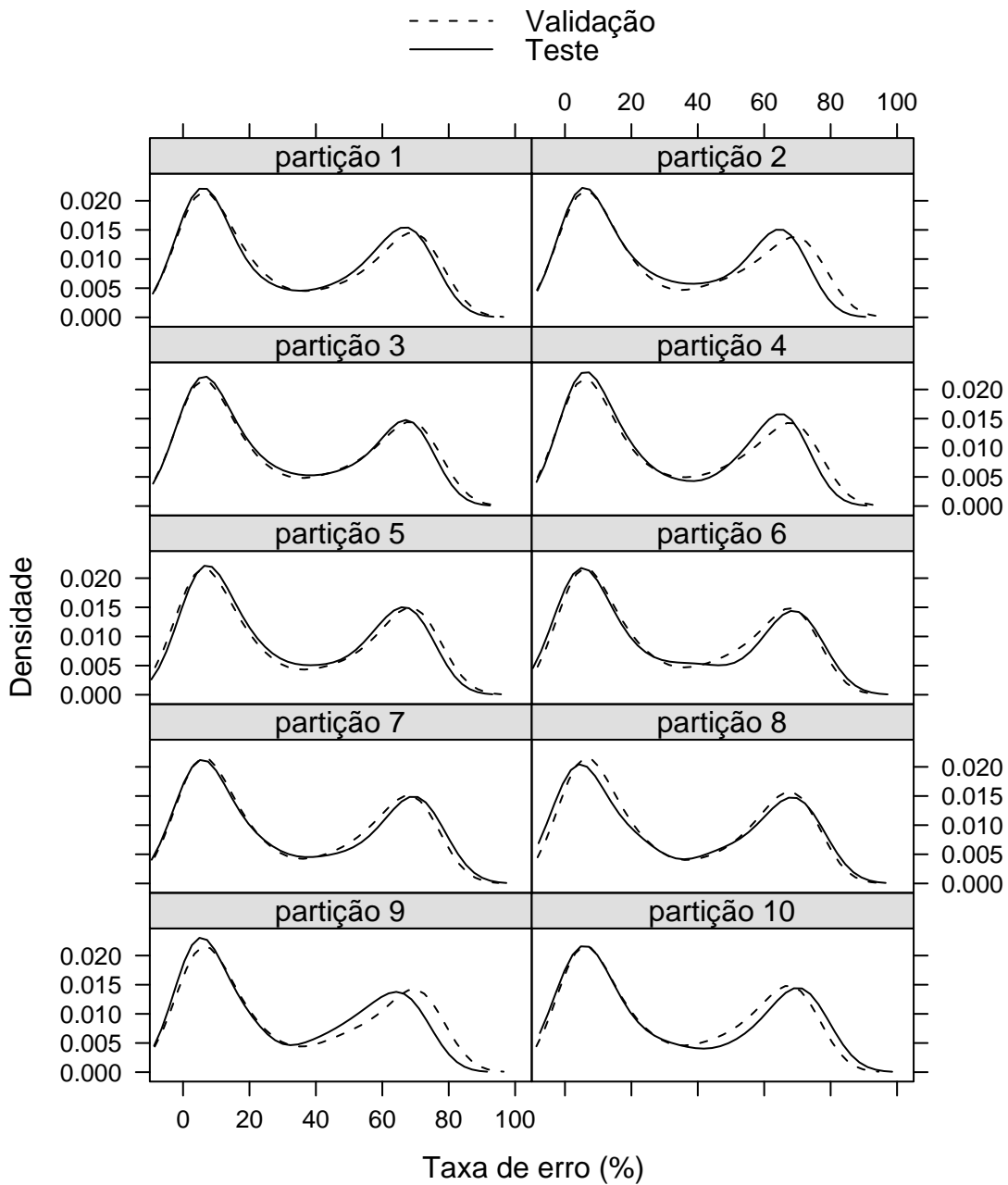


Figura A.13: Densidade da taxa de erro de validação e de teste para o conjunto de dados *segment* para todas as combinações de valores de parâmetros para SVMs.

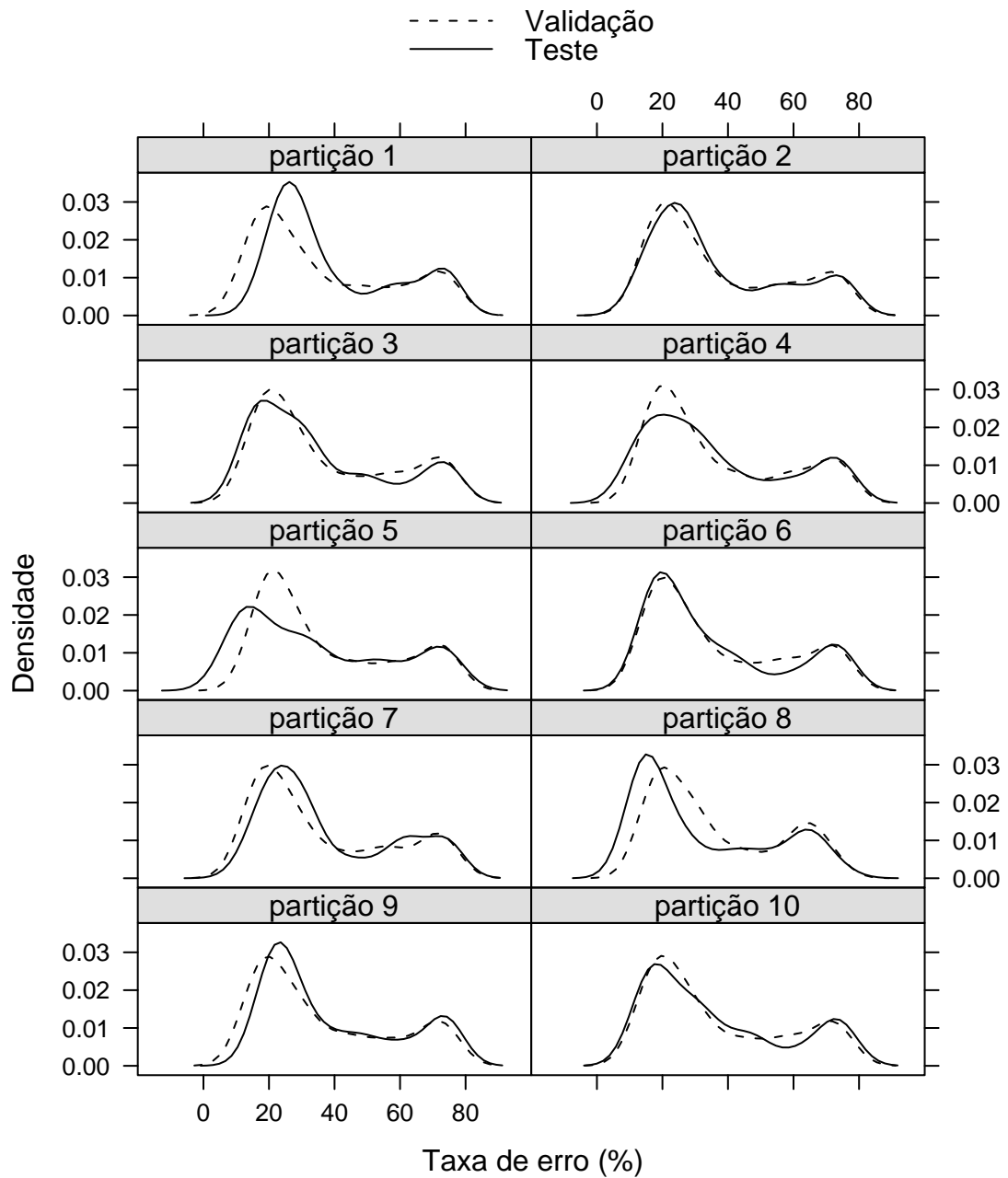


Figura A.14: Densidade da taxa de erro de validação e de teste para o conjunto de dados *vehicle* para todas as combinações de valores de parâmetros para SVMs.

Apêndice B

Curvas ROC

Neste apêndice são apresentados os gráficos de curva ROC para os conjuntos de dados binários utilizados nos experimentos da Seção 5.7. As figuras de B.1 a B.4 são referentes aos resultados obtidos nos experimentos com RNs, relatados na Seção 5.7.1. As figuras de B.5 a B.10 são referentes aos resultados obtidos nos experimentos com SVMs, reportados na Seção 5.7.2.

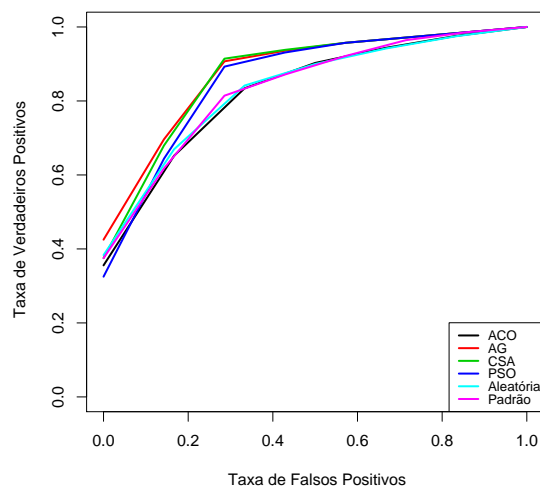


Figura B.1: Curvas ROC para o conjunto de dados Cólón usando RNs com parâmetros ajustados.

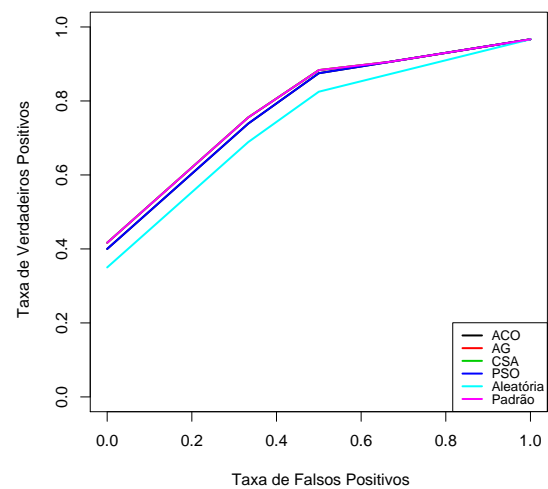


Figura B.2: Curvas ROC para o conjunto de dados Glioma usando RNs com parâmetros ajustados.

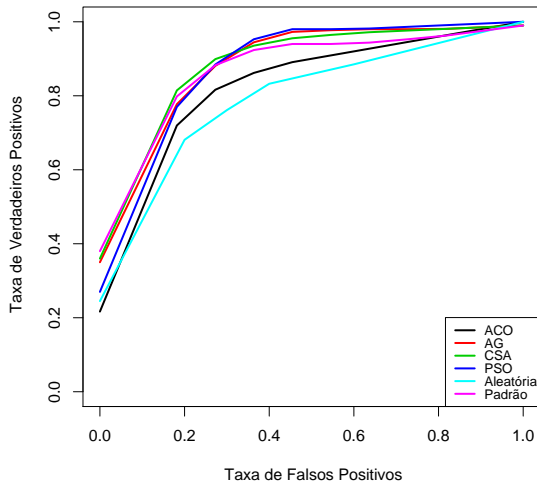


Figura B.3: Curvas ROC para o conjunto de dados Leucemia usando RNs com parâmetros ajustados.

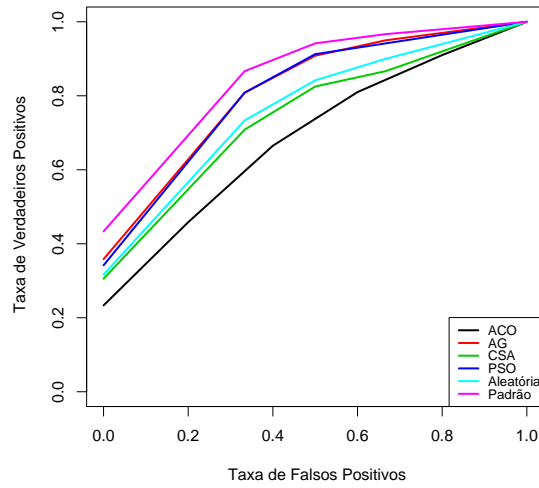


Figura B.4: Curvas ROC para o conjunto de dados Pâncreas usando RNs com parâmetros ajustados.

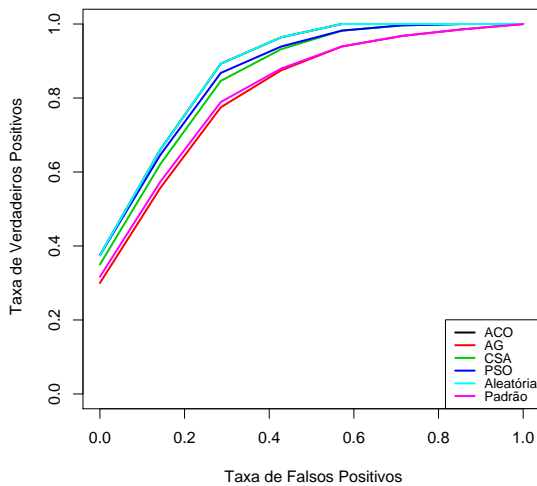


Figura B.5: Curvas ROC para o conjunto de dados Cólón usando SVMs com parâmetros ajustados.

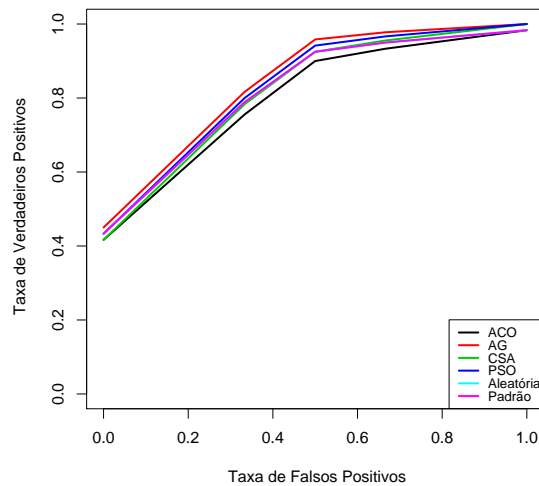


Figura B.6: Curvas ROC para o conjunto de dados Glioma usando SVMs com parâmetros ajustados.

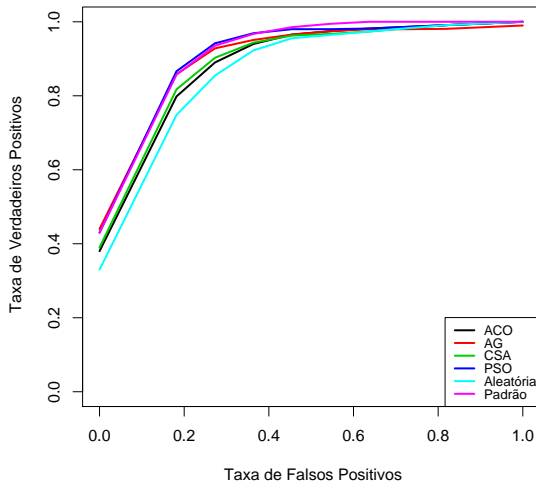


Figura B.7: Curvas ROC para o conjunto de dados Leucemia usando SVMs com parâmetros ajustados.

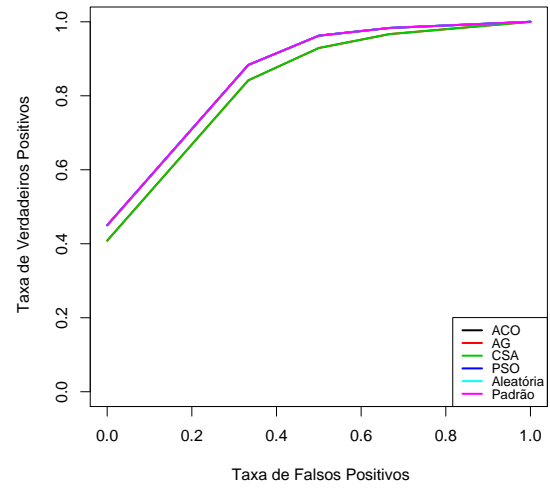


Figura B.8: Curvas ROC para o conjunto de dados Pâncreas usando SVMs com parâmetros ajustados.

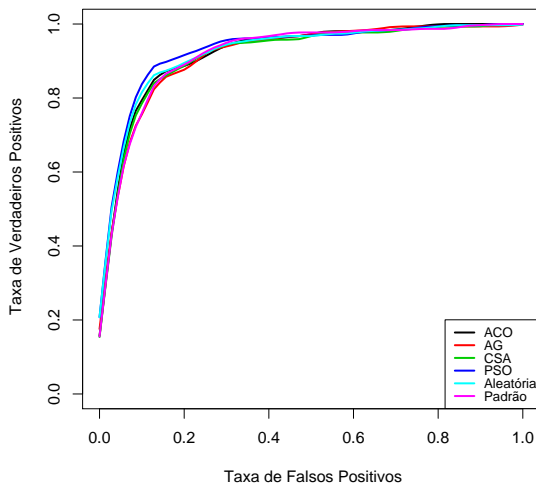


Figura B.9: Curvas ROC para o conjunto de dados *australian* usando SVMs com parâmetros ajustados.

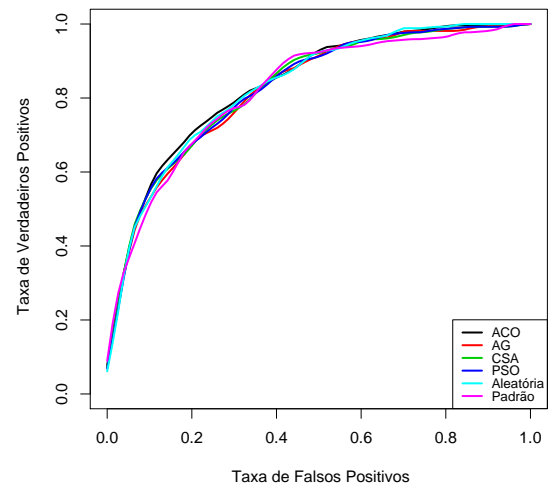


Figura B.10: Curvas ROC para o conjunto de dados *pima* usando SVMs com parâmetros ajustados.