



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES

Danilo Avilar Silva

**NOVAS ABORDAGENS DE PODA EM MÁQUINAS DE VETORES-SUORTE POR
MÍNIMOS QUADRADOS VIA COMPUTAÇÃO EVOLUTIVA**

Fortaleza - Ceará
25 de Agosto de 2015

Danilo Avilar Silva

**NOVAS ABORDAGENS DE PODA EM M ÁQUINAS DE
VETORES-SUPORTE POR M ÍNIMOS QUADRADOS
VIA COMPUTAC,ÃO EVOLUTIVA**

Dissertação submetida à Coordenação
do Programa de Pós-Graduação em
Engenharia de Telecomunicações do
Instituto Federal do Ceará, como requisito
parcial para obtenção do grau de Mestre
em Engenharia de Telecomunicações.

Área de concentração: Sistemas de
Telecomunicações

Linha de pesquisa: Processamento Digital
de Sinais

Orientador: Prof. Dr. Ajalmar Régio da
Rocha Neto

Fortaleza
2015

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

- S586n Silva, Danilo Avilar.
Novas abordagens de poda em máquinas de vetores-suporte por mínimos quadrados via computação evolutiva / Danilo Avilar Silva. - 2015.
126 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Engenharia de Telecomunicações, Campus Fortaleza, 2015.
Orientação: Prof. Dr. Ajalmar Rêgo da Rocha Neto.
1. Máquinas de Vetores-Suporte por Mínimos Quadrados. 2. Otimização Multiobjetivo. 3. Métodos de Poda. 4. Algoritmos Genéticos. 5. MOGAS LSSVM. I. Título.

CDD 621.382



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES

Danilo Avilar Silva

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Telecomunicações, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

Orientador: Prof. Dr. Ajalmar Rêgo da Rocha Neto
Instituto Federal do Ceará (IFCE)

Prof. Dr. João Paulo Pordeus Gomes
Universidade Federal do Ceará (UFC)

Prof. Dr. Regis Cristiano Pinheiro Marques
Instituto Federal do Ceará (IFCE)

Prof. Dr. Auzuir Ripardo de Alexandria
Instituto Federal do Ceará (IFCE)

Prof. Dr. Francisco José Alves de Aquino
Instituto Federal do Ceará (IFCE)

Fortaleza, 25 de Agosto de 2015

Dedico este trabalho aos meus pais Antonia e Manoel e ao meu irmão Daniel, minha amada família, a base de tudo!

Agradecimentos

Agradeço primeiramente a Deus pelo dom da vida e por me dar a sabedoria e a força necessária para enfrentar os obstáculos que a vida nos proporciona.

Muitas pessoas contribuíram para a realização deste trabalho, seja de forma direta ou indireta, seja demonstrando claramente o apoio ou ali no seu silêncio desejando que tudo desse certo. Enquanto para uns o término de uma conquista como essa significa o fim de um ciclo, para mim representa o início de um aprendizado contínuo.

Gostaria de agradecer aos meus pais Manoel e Antonia pela formação, carinho, conselhos e por serem exemplos de dedicação aos filhos. Estendo estes agradecimentos ao meu querido irmão Daniel que sempre me apoiou nos estudos. Muito obrigado por serem a base de tudo na minha vida, pela compreensão nos momentos difíceis, pelo incentivo e apoio em todas as decisões da minha vida. Obrigado por compreenderem a minha ausência durante este período e por sempre acreditarem que posso mais.

Agradeço a todos da família (não citarei nomes para não ser injusto e esquecer alguém, o que não é bem novidade vindo de mim) que mesmo distantes estamos de alguma forma sempre unidos. Muito obrigado à minha noiva Vilma pelo apoio incondicional às minhas decisões, por acreditar em mim, pela compreensão e incentivo na realização desta conquista.

O meu muito obrigado ao meu orientador professor Ajalmar com o qual ao longo da realização deste trabalho pude construir uma relação de amizade e respeito e, por toda a sua dedicação, crucial na elaboração desta dissertação e dos trabalhos que publicamos. Obrigado por aceitar este desafio e acreditar no meu potencial.

Gostaria ainda de agradecer:

- Aos meus amigos por compreenderem os momentos de ausência;
- Aos amigos guerreiros e companheiros de mestrado com os quais tive a oportunidade de aprender e quem sabe talvez ensinar;
- Ao time de pesquisa e desenvolvimento da empresa FOTOSensores ao qual faço parte, bem como aos gestores e diretores, que compreenderam a importância da realização desse

trabalho e me permitiram conciliar a vida profissional e acadêmica;

- Ao corpo docente do programa, em especial aos professores José Wally, Daniel Xavier e Francisco José com os quais tive a oportunidade de trabalhar de forma mais direta;
- Aos demais professores do IFCE pela formação acadêmica que tive na graduação e no mestrado, e aos demais servidores do PPGET e da instituição pela colaboração.

*“A menos que modifiquemos a nossa maneira de pensar,
não seremos capazes de resolver os problemas causados
pela forma como nos acostumamos a ver o mundo.”*

Albert Einstein

Resumo

A utilização das Máquinas de Vetores-Suporte por Mínimos Quadrados (LSSVM) como alternativa às Máquinas de Vetores-Suporte (SVM) é interessante devido a diminuição do alto custo computacional verificado no processo de treinamento das SVMs. Além disso, em geral, o desempenho obtido é equivalente à formulação da SVM padrão. Entretanto a falta de esparsidade presente nos classificadores LSSVM motiva a pesquisa e proposição de novos métodos para obtenção de conjuntos reduzidos e técnicas de poda para redução da quantidade de vetores-suporte.

A fim de combinar as vantagens do classificador LSSVM padrão e de Algoritmos Genéticos, são propostos neste trabalho, dois novos modelos de classificação capazes de manter e até melhorar o desempenho outrora obtido pelo classificador convencional. Paralelamente, tais modelos são capazes de obter significativas reduções em termos de vetores-suporte utilizados na obtenção do hiperplano ótimo de separação entre as classes.

A primeira proposta utiliza Algoritmos Genéticos Multiobjetivos com base em escalarização que levam em consideração tanto a acurácia quanto a redução de vetores-suporte. O compromisso entre os objetivos é definido por meio de um certo custo de poda presente na função de aptidão. Na segunda proposta, faz-se uso de Algoritmos Genéticos Mono-Objetivo que têm por base a acurácia obtida pelo classificador. Diferentemente da proposta anterior este modelo consiste em um classificador de tamanho fixo em que através de um parâmetro declarado *a priori* define-se o percentual de remoção de vetores-suporte. Dessa forma, os Algoritmos Genéticos trabalham para selecionar um conjunto de tamanho fixo de genes que melhor represente o conjunto de vetores-suporte.

As simulações computacionais realizadas com os classificadores propostos atestam um desempenho superior em termos de redução de vetores-suporte e valores equiparáveis de acurácia quando comparados à classificadores convencionais. Ao decorrer deste trabalho, também foi sugerida uma nova forma para manter fixa a quantidade de elementos em um indivíduo (uma solução para o sistema linear) com codificação binária submetido a um operador de reprodução.

Palavras-Chave: *Máquinas de Vetores-Suporte por Mínimos Quadrados, Otimização Multiobjetivo, Métodos de Poda, Algoritmos Genéticos, MOGAS-LSSVM, FSGAS-LSSVM.*

Abstract

The usage of Least Square Support-Vector Machines (LSSVMs) instead of Support-Vector Machine (SVMs) is very interesting since there is a reduction of computational cost for training models. Besides that, in general, LSSVM models achieve equivalent performance when compared to the conventional SVM ones. However, the absence of sparseness in the LSSVM classifiers motivates research and proposals of new methods for obtaining reduced sets and pruning methods in order to reduce the number of support vectors (SV).

In order to combine the advantages of standard LSSVM classifier and Genetic Algorithms (GAs), in this work is proposed two new models of classification which are able to maintain (or even to improve) the performance while to obtain significant reductions of support vectors. These support vectors are used to find the optimal hyperplane for separation between classes.

The first proposal uses Multi-Objective GAs scalarization-based that considers the accuracy and the reduction of support-vectors. The trade-off between the accuracy and pruning is defined by a certain cost of removal of this scalarization-based fitness function. This scalarization is understood as the process of transformation multi-objectives into only one objective. In the second proposal, also based on GAs, is used a mono-objective function which relies on the accuracy obtained by the classifier. Differently from the last proposal, this model consists in a fixed-size classifier that through a certain parameter defines the pruning percentage of support vectors. Thus, the GAs work to select a fixed-size group of genes (patterns) that represents in a better way the set of support vectors.

From the results, it is shown the robustness and efficiency of the proposed classifiers. During this work, it was also proposed a new way to maintain a fixed amount of genes with value equals in a binary-encoding individual (i.e., the number of support vectors in the solution for the LSSVM linear system) submitted to a reproduction operator.

Keywords: *Least Squares Support-Vector Machines, Multi-Objective Optimization, Pruning Methods, Genetic Algorithms, MOGAS-LSSVM, FSGAS-LSSVM.*

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

Lista de Abreviações

1	Introdução	22
1.1	Motivação	22
1.2	Objetivos Geral e Específicos	26
1.3	Produção Científica	26
1.4	Organização da Dissertação	27
2	Máquinas de Vetores-Suporte	29
2.1	Introdução	29
2.2	Conceitos Preliminares	30
2.3	Fundamentos Teóricos do Classificador SVM	34
2.3.1	Variáveis de Folga	34
2.3.2	Formulação Primal-Dual	35
2.4	Fundamentos Teóricos do Classificador LSSVM	39
2.4.1	Variáveis de Folga	39
2.4.2	Formulação Primal	41
2.5	Obtenção dos Parâmetros Ótimos	44

2.5.1	Matriz Inversa	44
2.5.2	Matriz Pseudo-Inversa	45
2.5.3	Otimização dos Hiperparâmetros	45
2.6	Conclusão	46
3	Algoritmos Genéticos	48
3.1	Introdução	48
3.2	Algoritmos Genéticos	50
3.2.1	Métodos de Seleção	53
3.2.2	Operadores de Cruzamento (ou Reprodução)	56
3.2.3	Operadores de Mutação	60
3.3	Otimização Mono e Multiobjetivo com Algoritmos Evolutivos	61
3.3.1	Hierarquia de Técnicas	62
3.3.2	Otimização Multiobjetivo com base em Escalarização	63
3.4	Conclusão	64
4	Métodos para Redução de Vetores-Suporte	66
4.1	Introdução	66
4.2	Classificadores Esparsos	68
4.2.1	Pruning LSSVM	70
4.2.2	IP-LSSVM	71
4.3	Classificadores Esparsos com base em AGs	73
4.3.1	GAS-LSSVM	73
4.3.2	MOGAS-LSSVM (Proposta 1)	75
4.3.2.1	Indivíduos (ou cromossomos)	75
4.3.2.2	Função de Aptidão	77
4.3.2.3	Parâmetros do Classificador MOGAS-LSSVM	77

4.3.2.4	Algoritmo MOGAS-LSSVM	78
4.3.3	FSGAS-LSSVM (Proposta 2)	78
4.3.3.1	Indivíduos (ou cromossomos)	79
4.3.3.2	Reprodução	80
4.3.3.3	Mutação	80
4.3.3.4	Função de Aptidão	81
4.3.3.5	Parâmetros do Classificador FSGAS-LSSVM	82
4.3.3.6	Algoritmo FSGAS-LSSVM	82
4.4	Conclusão	83
5	Simulações Computacionais	85
5.1	Metodologia	85
5.2	Resultados	88
5.2.1	Resultados para o Classificador MOGAS-LSSVM (Proposta 1)	89
5.2.2	Resultados para o Classificador FSGAS-LSSVM (Proposta 2)	94
5.2.3	Comparação entre os classificadores	100
5.3	Conclusão	113
6	Conclusões e Trabalhos Futuros	114
6.1	Conclusões	114
6.2	Contribuições deste Trabalho	116
6.3	Trabalhos Futuros	116
	Bibliografia	118

Lista de Figuras

1	Mapeamento não linear do espaço de dados de entrada para o espaço de características de alta dimensão.	31
2	Interpretação geométrica da distância de um vetor $\mathbf{x}_i \in \{\pm 1\}$ ao hiperplano ótimo $H_o^{(s)}$ e da margem de separação ρ	32
3	Divisão do espaço de características do classificador SVM.	34
4	Interpretação geométrica das variáveis de folga em classificadores SVM.	35
5	Divisão do espaço de características do classificador LSSVM.	39
6	Análise das variáveis de folga utilizadas no classificador LSSVM.	40
7	Diagrama básico de funcionamento de um algoritmo genético.	50
8	Neste método de seleção, Ind_T indivíduos ($Ind_T \geq 2$) da população são escolhidos aleatoriamente com igual probabilidade. Em seguida, é realizado um torneio no qual o indivíduo mais apto é selecionado.	55
9	Comparação entre as porções destinada a indivíduos em uma roleta antes e após a consideração do valor de adequação. (a) distribuição da roleta em função do valor de aptidão; (b) distribuição da roleta após a consideração do valor de adequação.	56
10	Processo de reprodução através do operador de cruzamento de único ponto.	58
11	Processo de reprodução através do operador de cruzamento multiponto.	58
12	Demonstração do procedimento de geração de filhos por Ponto Uniforme: (a) bits da máscara iguais a 0 referenciam genes do <i>Pai 1</i> , caso contrário, o material genético em questão será do <i>Pai 2</i> ; (b) bits da máscara iguais a 0 referenciam genes do <i>Pai 2</i> , caso contrário, o material genético em questão será do <i>Pai 1</i>	58

13	Processo de cruzamento baseado em OX: (a) pais selecionados; (b) escolha aleatória do subconjunto de genes para mapeamento dos filhos; (c) criação do cromossomo protótipo com material genético do Pai_2 exceto para valores iguais ao subconjunto S_{P_1} ; (d) deslocamento de bits a esquerda para preenchimento dos campos do 2º segmento; (e) cópia do subconjunto S_{P_1} para o protótipo e geração do novo filho.	59
14	Funcionamento do operador de mutação Flip: (a) cromossomo escolhido para mutação; (b) escolha aleatória do gene a ser mutado em função da probabilidade de mutação; (c) cromossomo mutado.	60
15	Funcionamento do operador de mutação Swap: (a) cromossomo escolhido para mutação; (b) escolha aleatória do(s) par(es) de troca; (c) cromossomo mutado.	60
16	Funcionamento do operador de mutação Creep: (a) cromossomo escolhido para mutação; (b) operação de adição sob o valor do gene escolhido aleatoriamente para ser mutado; (c) cromossomo mutado.	61
17	Hierarquia de técnicas existentes em Otimização Multiobjetivo. Traduzido de Veldhuizen e Lamont (2000).	63
18	Processo de transformação entre codificação binária-inteira para aplicação de operador de cruzamento OX1 seguido pela transformação entre codificação inteira-binária.	81
19	Esquema que descreve o procedimento de mutação onde garante-se a quantidade de genes definidos como “um” constante.	82
20	Diagrama de blocos da metodologia utilizada.	88
21	Comparação de desempenho entre os classificadores MOGAS-LSSVM, GAS-LSSVM e LSSVM para o problema VCP. (a) Acurácia <i>versus</i> variação do tamanho do conjunto de treinamento. (b) Média de VS <i>versus</i> variação do tamanho do conjunto de treinamento.	91
22	Desempenho do classificador MOGAS-LSSVM em função do custo de poda para o problema VCP: (a) Acurácia <i>versus</i> diferentes valores de tamanho do conjunto de treinamento em função do θ no intervalo de $[0,1]$. (b) Média de vetores-suporte <i>versus</i> diferentes valores de tamanho do conjunto de treinamento em função do θ no intervalo de $[0,1]$	92

23	Acurácia e redução em função do custo de poda θ , obtidos para cada problema e avaliados com 80% e 20% do conjunto de treinamento e teste, respectivamente.	92
24	Acurácia e redução em função do custo de poda θ obtidos para cada problema avaliados com 60%, 20% e 20% do conjunto de treinamento, validação e teste, respectivamente.	93
25	Desempenho do classificador FSGAS-LSSVM para o problema VCP em função do percentual de poda de vetores-suporte previamente definido: (a) Acurácia <i>versus</i> variação do tamanho do conjunto de treinamento. (b) Média de vetores-suporte <i>versus</i> variação do tamanho do conjunto de treinamento.	97
26	Comparação entre acurácia dos métodos FSGAS-LSSVM e IP-LSSVM em função do percentual de remoção de vetores-suporte para cada problema avaliado (<i>Abordagem I</i>).	98
27	Comparação entre acurácia dos métodos FSGAS-LSSVM e P-LSSVM em função do percentual de remoção de vetores-suporte para cada problema avaliado (<i>Abordagem II</i>).	98
28	Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema VCP. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.	100
29	Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema HAB. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.	103
30	Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema BCW. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.	105

- 31 Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema PID. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%. 107
- 32 Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema RIP. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%. 109
- 33 Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema BAN. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%. 111

Lista de Tabelas

1	Exemplos de <i>kernels</i> tipicamente utilizados.	38
2	Exemplos de representação de indivíduos.	52
3	Atribuição do valor de adequação para indivíduos no método de Seleção por torneio.	56
4	Características dos conjuntos de dados utilizados neste trabalho.	86
5	Resumo da metodologia abordada.	87
6	Resultados para os classificadores LSSVM, IP-LSSVM, GAS-LSSVM e MOGAS-LSSVM com 80% (20%) do conjunto de dados para treinamento (teste).	89
7	Resultados para os classificadores LSSVM, P-LSSVM, GAS-LSSVM e MOGAS-LSSVM com 60%, 20% e 20% do conjunto de dados para treinamento, validação e teste, respectivamente.	90
8	Tempo médio de treinamento para os classificadores IP-LSSVM (<i>Abordagem I</i>), P-LSSVM (<i>Abordagem II</i>), LSSVM e MOGAS-LSSVM (<i>Abordagens I e II</i>).	94
9	Resultados para os classificadores LSSVM, IP-LSSVM e FSGAS-LSSVM com 80% (20%) do conjunto de dados de treinamento (teste).	95
10	Resultados para os classificadores LSSVM, P-LSSVM e FSGAS-LSSVM com 60%, 20% e 20% do conjunto de dados para treinamento, validação e teste, respectivamente.	96
11	Tempo médio de treinamento para os classificadores IP-LSSVM (<i>Abordagem I</i>), P-LSSVM (<i>Abordagem II</i>), LSSVM e FSGAS-LSSVM (<i>Abordagens I e II</i>).	99
12	Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% até 80% do conjunto de dados para treinamento considerando o problema VCP ($\nu = 0,05$).	102

13	Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% at é 80% do conjunto de dados para treinamento considerando o problema HAB ($\gamma = 0,04$).	104
14	Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% at é 80% do conjunto de dados para treinamento considerando o problema BCW ($\gamma = 0,04$).	106
15	Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% at é 80% do conjunto de dados para treinamento considerando o problema PID ($\gamma = 0,04$).	108
16	Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% at é 80% do conjunto de dados para treinamento considerando o problema RIP ($\gamma = 0,04$).	110
17	Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% at é 80% do conjunto de dados para treinamento considerando o problema BAN ($\gamma = 0,04$).	112

Lista de Símbolos

$\mathbf{x}_i$	Vetor de atributos de entrada
y_i	Saída do classificador
H_o	Hiperplano
$\mathbf{w}$	Vetor de pesos do hiperplano
$\phi(\cdot)$	Mapeamento não linear
b	Viés do hiperplano
$\mathbb{R}$	Conjunto dos números reais
$\mathbb{R}^n$	Conjunto dos números reais n -dimensional
$\mathbb{R}^{n_h}$	Conjunto dos números reais em um espaço de alta dimensão
N	Quantidade de elementos do conjunto de treinamento
J_P	Função custo do classificador LSSVM
H_{+1}	Hiperplano da classe +1
H_{-1}	Hiperplano da classe -1
$P(x_a, y_a)$	Coordenadas do ponto P
r	Representação da reta $ax_a + by_a + c = 0$ sobre o plano
d	Distância do ponto P a reta r
d_i	Distância do vetor $\mathbf{x}_i$ ao hiperplano H_o
$\ \mathbf{w}\ _k$	Norma euclidiana do vetor peso
$H_o^{(s)}$	Hiperplano ótimo de separação
ρ	Margem de separação

d_i^{+1}	Mínima distância entre um padrão $\in \{+1\}$ e o hiperplano $H_o^{(s)}$
d_i^{-1}	Mínima distância entre um padrão $\in \{-1\}$ e o hiperplano $H_o^{(s)}$
ξ	Variável de folga
ν	Termo de Regularização
β, α	Multiplicador de Lagrange
$K(,)$	Função de Kernel
σ	Desvio Padrão
$sign()$	Função Sinal
$\mathbf{x}^+$	Padrão $\mathbf{x}$ pertencente a classe $y_i = +1$
$\mathbf{x}^-$	Padrão $\mathbf{x}$ pertencente a classe $y_i = -1$
$f_r(i)$	Valor de aptidão relativa do i -ésimo indivíduo
$f_a(i)$	Valor de aptidão absoluta do i -ésimo indivíduo
M	Tamanho da população
t	Geração
p_q	Ponto de separação em operadores de cruzamento
$P(t)$	População da geração t
θ	Custo de poda do classificador MOGAS-LSSVM

Lista de Abreviações

AG	Algoritmos Genéticos
BMU	Best Matching Unit
CE	Computação Evolutiva
CX	Cyclic Crossover
EE	Estratégias Evolutivas
ERM	Empirical Risk Minimization
FSGAS-LSSVM	Fixed-Size Genetic Algorithm for Sparse LSSVM
GAS-LSSVM	Genetic Algorithm for Sparse LSSVM
KKT	Karush-Kuhn-Tucker
LSSVM	Least Square Support-Vector Machines
MD	Marcador de Decisão
MLP	Multilayer Perceptron
MOGA	Multi-Objective Genetic Algorithms
MOGAS-LSSVM	Multi-Objective Genetic Algorithm for Sparse LSSVM
OX	Order Crossover
PE	Programação Evolutiva
PG	Programação Genética
PMX	Partially Matched Crossover
QP	Programação Quadrática
RBF	Radial Basis Function

RNA	Redes Neurais Artificias
SMO	Sequential Minimal Optimization
SRM	Structural Risk Minimization
SVM	Support-Vector Machines
VC	Dimensão <i>Vapnik-Chervonenkis</i>
VS	Vetores-Suporte

INTRODUÇÃO

Reconhecimento de padrões é uma área cujo objetivo consiste na associação de objetos (ou padrões) a um certo número de categorias (ou classes). Estes padrões estão diretamente interligados ao tipo de problema analisado e, assim, podem representar imagens, sinais que expressam determinadas formas de onda ou até resultados de medições que necessitam de classificação (THEODORIDIS; KOUTROUMBAS, 2008).

Até a década de 60, pesquisas teóricas voltadas ao reconhecimento de padrões na área da estatística foram o foco de estudo de pesquisadores (THEODORIDIS; KOUTROUMBAS, 2008). Contudo, a necessidade de aplicações para classificação de padrões estimulada pelo advento dos computadores, permitiu a evolução de pesquisas em aplicações mais complexas nas mais diversas áreas como: bioinformática, reconhecimento biomédico, análise de documentos por imagens para reconhecimento de caracteres, mineração de dados, entre outras (PADILHA, 2013).

Abordagens dinâmicas como Redes Neurais Artificiais (*Artificial Neural Network, RNA*) são capazes de ajustar seus pesos adequando-os às características dos dados fornecidos sem um conhecimento mais detalhado acerca do problema. Isto é possível pois seu funcionamento tem por base um processo de treinamento iterativo, onde os pesos são atualizados a cada etapa. Apesar de redes neurais serem amplamente aplicadas em problemas de classificação de padrões, o uso de Máquinas de Vetores-Suporte (*Support Vector-Machines, SVM*) ganhou grande destaque por fornecerem uma superfície ótima de separação entre classes (PADILHA, 2013).

1.1 Motivação

Uma SVM é uma generalização não linear do algoritmo proposto por Vapnik e Lerner (1963), denominado *Generalized Portrait*. Posteriormente, foi proposta uma descrição do algoritmo para a construção de hiperplanos de separação com margem ótima (BOSER;

GUYON; VAPNIK, 1992). Após décadas de pesquisas e com um forte embasamento na teoria de aprendizagem estatística (VAPNIK, 1995, 1998) surgem as SVMs (BURBIDGE; BUXTON, 2001).

Conceitualmente, o funcionamento das SVMs se resume no mapeamento de vetores de entrada para um espaço de alta dimensão ($\mathbb{R}^n \rightarrow \mathbb{R}^h$). Nesse espaço, as características para uma superfície de decisão linear são encontradas (CORTES; VAPNIK, 1995). É considerado um método de aprendizagem supervisionado aplicado tanto a problemas de classificação como de regressão. A SVM baseia-se no princípio de minimização do risco estrutural (*Structural Risk Minimization, SRM*) que busca a maximização da capacidade de generalização, o que resulta em modelos com desempenho superior às redes neurais convencionais que empregam apenas a minimização do risco empírico (*Empirical Risk Minimization, ERM*) (GUNN, 1998).

De um modo geral, o resultado do mapeamento realizado no processo de treinamento da SVM consiste em encontrar os padrões mais próximos ao hiperplano de separação entre as classes. Esses padrões são denominados de vetores-suporte (VS) e representam os vetores mais informativos do conjunto de treinamento. Dessa forma, os vetores-suporte são os únicos vetores a serem armazenados para posterior apresentação e avaliação de padrões não vistos na etapa de teste ¹.

Um fator limitante ao uso do classificador SVM é a complexidade de resolução (CARVALHO, 2005), uma vez que treinar uma Máquina de Vetores-Suporte exige a resolução de um problema de programação quadrática (QP) com restrições de desigualdade (OSUNA; FREUND; GIROSI, 1997; PLATT, 1998). Outro fator importante para sua devida utilização, consiste na escolha adequada da função de kernel e, consequentemente, da escolha dos parâmetros associados a ela. Esta função é responsável por elevar a dimensão dos dados utilizados para que seja possível separá-los através de um hiperplano ².

Com o propósito de simplificar o processo de obtenção dos multiplicadores de Lagrange da SVM ³, formulou-se originalmente um classificador para problemas de duas classes denominado LSSVM. Tal modelo considera o uso de restrições de igualdade em vez de desigualdade, bem como, a utilização de variáveis de folga ao quadrado na função custo. Como resultado dessas modificações, a solução do problema se resume à resolução de um sistema de equações lineares ao invés da resolução de um problema QP (SUYKENS; VANDEWALLE, 1999a).

¹ Vale ressaltar que também devem ser armazenados os valores dos multiplicadores de Lagrange.

² Tal característica está associada a qualquer classificador que utilize uma função de kernel.

³ Variáveis auxiliares inseridas durante o processo de resolução da formulação primal de otimização.

Em geral, o custo computacional e a complexidade para resolução de um sistema de equações lineares é menor do que a resolução de um problema de programação quadrática. Assim, uma LSSVM apresenta-se como uma excelente alternativa em virtude do baixo custo computacional para treinamento, e que, em geral, mantém a mesma qualidade nas soluções, quando comparada com uma SVM para um mesmo problema.

Em contrapartida, essas modificações fazem com que a LSSVM perca a detecção automática dos vetores-suporte, uma das características principais presente na SVM. Isso ocorre devido a falta de esparsidade da solução obtida, dessa forma todos os padrões do conjunto de treinamento são considerados vetores-suporte. Como dito anteriormente, são os vetores-suporte que possibilitam a construção da superfície de separação para a correta classificação dos dados na etapa de teste.

Devido a essas características, foram iniciadas pesquisas com o foco na redução do conjunto de vetores-suporte a fim de obter apenas os vetores necessários para a determinação da fronteira de decisão. Tais pesquisas resultaram em técnicas e métodos que normalmente são denominados de Métodos de Poda ou Métodos de Obtenção de Conjuntos Reduzidos ⁴ introduzidos por Burges (1996). Nesse trabalho, é apresentado um método que calcula uma aproximação para a regra de decisão do classificador SVM com um número muito menor em termos de vetores-suporte, que pode ser utilizado tanto para problemas de reconhecimento de padrões como para problemas de regressão de funções. O conjunto reduzido determinado pelo método de Burges não é necessariamente composto por vetores-suporte.

Como resultado, observa-se uma melhoria significativa em termos de velocidade de classificação sem grandes efeitos sobre a acurácia quando o conjunto de testes é avaliado utilizando um kernel polinomial de segundo ou terceiro graus. O método de Burges foi refinado em (BURGES; SCHOLKOPF, 1997), entretanto, a determinação do conjunto reduzido provou ter um alto custo computacional, além de poder apresentar mínimos locais (HADY *et al.*, 2011). O conjunto reduzido de Burges conseguiu acelerar o processo de classificação sem afetar significativamente a capacidade de generalização, contudo, devido aos problemas já citados, tal método não ganhou maior destaque em termos de utilização (DOWNS; GATES; MASTERS, 2001).

Em termos de redução de vetores-suporte aplicado à LSSVM, Suykens, Lukas e Vandewalle (2000a) propuseram um método que, apesar da norma de mínimos quadrados intrínseca ao modelo, impõe esparsidade com base em um espectro ordenado de valores de suporte. De certa

⁴ Esse termo também é utilizado para métodos que visam reduzir o conjunto de vetores-suporte em modelos SVM.

⁵ Multiplicadores de Lagrange.

forma, pode-se afirmar que este é um dos métodos mais importantes utilizado para obtenção de classificadores esparsos, tanto pela facilidade de implementação como pelo desempenho obtido nos resultados.

Mesmo obtendo vetores com valores de suporte não-nulos, tais vetores indicam a importância relativa de como cada um dos pontos de dados contribuem para o modelo. Um espectro de valores pode ser observado através de uma representação gráfica ordenada dos valores absolutos dos multiplicadores de Lagrange (SUYKENS; LUKAS; VANDEWALLE, 2000a). Este espectro de valores de suporte é usado para obter esparsidade, impondo de forma gradual, vetores-suporte nulos para pontos de dados do conjunto de treinamento menos significativos. Após isso, o classificador é novamente treinado, com base no conjunto de treinamento reduzido, e vetores são eliminados a cada iteração até que um critério de parada seja alcançado, estando este, geralmente associado à diminuição de desempenho do conjunto de validação.

Posteriormente, vários outros métodos foram propostos com a mesma finalidade (LI; LIN; ZHANG, 2006; VALYON; HORVATH, 2004). Uma abordagem recente foca na solução para conjuntos de dados de grande escala por meio de tamanho fixo (MALL; SUYKENS, 2013). Já em Hoegaerts *et al.* (2004) é realizada uma comparação entre algoritmos de poda para LSSVM.

A escolha equivocada de parâmetros do kernel pode levar a um baixo desempenho do classificador. Tais escolhas são feitas geralmente por métodos iterativos como *grid search*, que requerem a determinação manual da região e resolução de pesquisa (GIJSBERTS; METTA; ROTHKRANTZ, 2010). Isto se faz necessário para que essa busca seja computacionalmente viável.

Com isso, o uso de ferramentas de Computação Evolutiva (CE), tais como Algoritmos Genéticos, tem contribuído significativamente nas mais diversas linhas de pesquisa para resolução de problemas de otimização. AGs são métodos de otimização que podem ser utilizados para gerar soluções úteis para problemas de busca. Sua utilização juntamente com modelos para tarefas de classificação, consiste no ajuste de parâmetros internos do classificador (HUANG; HU; YANG, 2011; SAMADZADEGAN; SOLEYMANI; ABBASPOUR, 2010; REN; BAI, 2010). Por outro lado, outros métodos de meta-heurística também têm sido utilizados com tal propósito (SANTOS *et al.*, 2012; AYDIN; KARAKOSE; AKIN, 2011; ZHANG; CHEN; HE, 2010).

Proposições desse tipo voltadas à escolha de parâmetros ótimos do kernel e do classificador LSSVM, podem ser encontradas em diversos outros trabalhos (DONG; LUO, 2013; MUSTAFA *et al.*, 2012; MOUSAVI; IRANMANESH, 2011).

Uma outra vertente pouco difundida na literatura atual, objeto de estudo e contribuição, desta dissertação, consiste na utilização de Algoritmos Genéticos com foco na redução do número de vetores-suporte de Máquinas de Vetores-Suporte por Mínimos Quadrados. O método em Silva e Rocha Neto (2013) tem como ideia principal excluir padrões não relevantes ou que podem estar corrompidos por ruído, garantindo ao classificador desempenho sem impactos negativos e reduzindo o número de vetores-suporte. Segundo ainda Silva e Rocha Neto (2013), este método pode ser também aplicado a problemas com grandes conjuntos de dados e mostra-se viável em aplicações que dispõem de recursos computacionais limitados.

1.2 Objetivos Geral e Específicos

Este trabalho tem como objetivo realizar um estudo comparativo entre métodos de poda com o intuito de obter conjuntos reduzidos de vetores-suporte em classificadores LSSVM. Adicionalmente, busca-se propor e avaliar a combinação de técnicas de Computação Evolutiva para a obtenção de conjuntos reduzidos.

Em seu sentido mais particular, esta dissertação tem os seguintes objetivos específicos:

- apresentar novos métodos para obtenção de conjuntos reduzidos em LSSVM para tarefas de classificação;
- propor e avaliar o desempenho do uso de técnicas de Computação Evolutiva na obtenção de conjuntos reduzidos;
- apresentar um classificador esparso com base em Algoritmos Genéticos Multiobjetivos;
- manter o desempenho do classificador, mesmo com um conjunto de vetores-suporte reduzido.

1.3 Produção Científica

Durante este projeto de mestrado, os seguintes trabalhos científicos foram aceitos e publicados, a saber:

- **SILVA, D.A.; SILVA, J.P.; ROCHA NETO, A.R. Novel approaches using evolutionary computation for sparse least square support vector machines.** Journal Elsevier Neurocomputing, 2015 - ISSN: 0925-2312. Disponível em: <<http://dx.doi.org/10.1016/j.neucom.2015.05.034>> .

- **SILVA, D.A.; ROCHA NETO, A.R. A genetic algorithms-based lssvm classifier for fixed-size set of support vectors.** In: International Work-Conference on Artificial Neural Networks - IWANN 2015. Palma de Mallorca, Spain: Springer International Publishing, 2015, (Lecture Notes in Computer Science, v. 9095). p. 127-141. Disponível em: <<http://link.springer.com/chapter/10.1007%2F978-3-319-19222-2.11>> .
- **SILVA, D.A.; ROCHA NETO, A.R. Multi-objective genetic algorithms for sparse least square support vector machines.** In: Intelligent Data Engineering and Automated Learning - IDEAL 2014. Salamanca, Spain: Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8669). p. 158-166. Disponível em: <<http://link.springer.com/chapter/10.1007%2F978-3-319-10840-7.20>> .

1.4 Organização da Dissertação

O restante desta dissertação é dividida em 5 capítulos: Máquinas de Vetores-Suporte; Algoritmos Genéticos; Métodos para Redução de Vetores-Suporte; Simulações Computacionais e, por fim, Conclusões e Trabalhos Futuros.

Em todos os capítulos busca-se detalhar o assunto abordado de tal forma que o leitor compreenda as considerações expostas e as contribuições desta dissertação, sempre associando as ideias apresentadas ao contexto presente na literatura.

No Capítulo 2, é descrita a fundamentação teórica tanto das Máquinas de Vetores-Suporte como das Máquinas de Vetores-Suporte por Mínimos Quadrados. Em seguida são apresentadas as principais diferenças dos classificadores LSSVM com relação à SVM. O nível de detalhes adotado ao decorrer do capítulo permitirá o correto entendimento do uso de técnicas de redução de complexidade apresentadas mais adiante.

Em virtude das propostas apresentadas no Capítulo 4 terem por base a utilização de Algoritmos Genéticos para a obtenção de um conjunto reduzido de vetores-suporte, será abordado no Capítulo 3 a fundamentação teórica necessária para o entendimento desse método. Aqui são apresentados também alguns métodos clássicos de seleção de indivíduos, operadores de reprodução e de mutação. Além disto, é apresentada a evolução dos indivíduos proporcionada pelos AGs.

Os métodos de redução de vetores-suporte utilizados como base de comparação com os métodos propostos são descritos no Capítulo 4. Outras técnicas relevantes encontradas na literatura também são apresentadas de forma sucinta. Pode-se dizer que as contribuições mais

significativas deste trabalho são apresentadas nesse capítulo.

Os métodos *Multi-Objective Genetic Algorithm for Sparse LSSVM (MOGAS-LSSVM)* e *Fixed-Size Genetic Algorithm for Sparse LSSVM (FSGAS-LSSVM)* são descritos. Ambos os métodos têm por base a utilização de Algoritmos Genéticos com o objetivo de maximizar a taxa de acerto do classificador e minimizar o número de vetores-suporte. Basicamente, a principal diferença entre os métodos consiste no processo de otimização. Em MOGAS-LSSVM, o processo de otimização é guiado pelo custo de poda, enquanto no FSGAS-LSSVM, o percentual de redução desejado é informado previamente.

Em seguida, no Capítulo 5, a metodologia utilizada, as simulações computacionais e os resultados obtidos são apresentados. Tabelas com os resultados obtidos para os métodos propostos e para os métodos usados para comparação, gráficos relacionando desempenhos obtidos e percentuais de redução quando os métodos são submetidos a diversas bases de dados retiradas do repositório UCI Machine Learning são expostos.

O Capítulo 6 descreve as considerações finais e uma conclusão geral, ressaltando as principais contribuições deixadas durante este trabalho de mestrado. Por fim, sugestões para trabalhos futuros são apresentadas.

MÁQUINAS DE VETORES -SUPORTE

Neste capítulo, as Máquinas de Vetores-Suporte e sua derivação por Mínimos Quadrados são os objetos de estudo. Será apresentada a formulação para a concepção dos classificadores SVM e LSSVM padrão. Pretende-se ainda, apresentar as motivações relacionadas a concepção das Máquinas de Vetores-Suporte por Mínimos Quadrados e as vantagens na utilização desse classificador em problemas de classificação de padrões.

Sempre que possível, comparações entre os classificadores LSSVM e SVM são realizadas, facilitando a compreensão sobre como a resolução de sistemas de equações lineares ao invés de problemas de programação quadrática superam o alto custo computacional de treinamento presente nos classificadores SVM.

É descrito também, como a falta de esparsidade no vetor de multiplicadores de Lagrange acomete os classificadores LSSVM. Isto ocorre devido modificações intrínsecas à formulação do modelo em relação a formulação da SVM padrão.

2.1 Introdução

A utilização de RNAs como modelos matemáticos com capacidade de aproximação universal resulta em algoritmos de treinamento com capacidade de generalização, porém ainda sujeitos ao problema de adequação do modelo utilizado aos dados de treinamento apresentados (*overfitting*). Conhecer a complexidade do problema a ser avaliado juntamente com suas não-linearidades nem sempre é possível. Isso torna a capacidade de generalização de modelos de otimização uma tarefa custosa, e que por vezes resulta em soluções não satisfatórias (FERRAO *et al.*, 2007; MARETTO, 2007).

Com isso, devido ao alto poder de generalização e com um forte embasamento estatístico, a utilização de Máquinas de Vetores-Suporte tornou-se uma excelente alternativa tanto para resolução de problemas de classificação de padrões como para problemas de aproximação

de funções. Neste método os dados de entrada são mapeados para um espaço de altíssima dimensão, de tal maneira que é possível a construção de um hiperplano de separação ótimo no espaço de características. Basicamente, para treinar um classificador SVM exige-se que um problema de otimização quadrática seja resolvido, enquanto métodos de treinamento com base em gradiente, além do problema descrito anteriormente, estão sujeitos a encontrar mínimos locais.

Apesar de evitar sobreajuste¹, apresentar robustez em grandes dimensões, ter sua teoria bem fundamentada na matemática e estatística e sua elevada capacidade de generalização, a resolução de problemas de otimização quadrática intrínseco à formulação da SVM pode ser computacionalmente muito custoso (MARETTO, 2007).

A pergunta em questão é: o quanto se pode simplificar a formulação das Máquinas de Vetores-Suporte sem perder suas principais vantagens? Diante disto, Suykens e Vandewalle (1999a) propuseram uma alternativa à SVM, denominada Máquinas de Vetores-Suporte por Mínimos Quadrados que foi originalmente proposta para resolver problemas binários, mas que através de uma combinação de classificadores pode também resolver problemas multiclasse (SUYKENS; VANDEWALLE, 1999b).

2.2 Conceitos Preliminares

O classificador SVM baseia-se em uma estimativa linear que aproxima a relação entre os pares de entrada $\mathbf{x}_i$ e saída y_i , $(\mathbf{x}_i, y_i)$ pela equação de um hiperplano:

$$H_o : y_i = \mathbf{w}^T \phi(\mathbf{x}_i) + b, \quad (2.1)$$

em que $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{\pm 1\}$, a partir de um conjunto de treinamento $(\mathbf{x}_i, y_i)_{i=1}^N$, de tal maneira que $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ é um mapeamento não linear realizado do espaço dos dados de entrada para um espaço de características de alta dimensão. Conforme ilustrado na Figura 1, através do mapeamento ϕ para um espaço de alta dimensão, busca-se traçar um hiperplano de separação linear entre as classes $\{\pm 1\}$. O viés b e o vetor de pesos $\mathbf{w}$ normal ao hiperplano H_o , são estimados pelo algoritmo de otimização que tem como objetivo a minimização de uma função custo a ser detalhada mais a frente.

Um padrão é considerado corretamente classificado caso sua localização esteja

¹ Situação em que o modelo se adequa aos dados utilizados no processo de treinamento e perde a capacidade de generalização, apresentando baixa acurácia sobre o conjunto de teste.

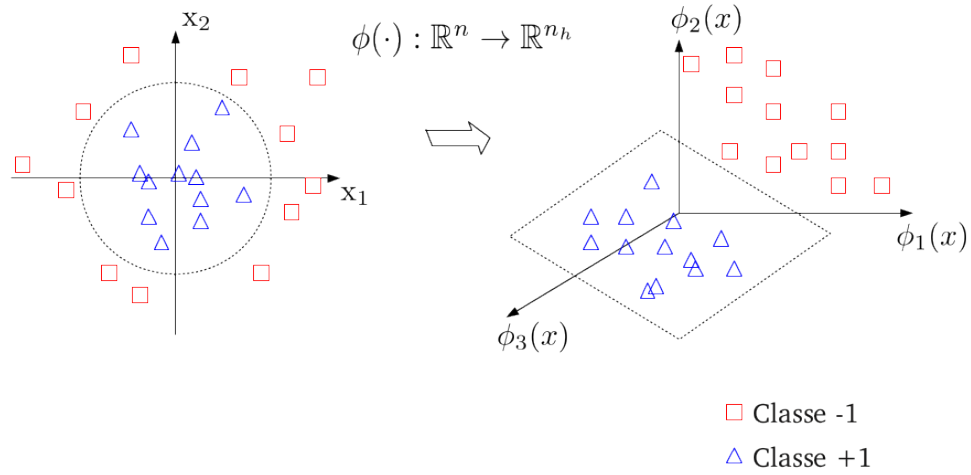


Figura 1: Mapeamento não linear do espaço de dados de entrada para o espaço de características de alta dimensão.

compreendida fora da margem de apoio da sua respectiva classe, isto é, quando

$$\begin{cases} H_{+1} : \mathbf{w}^T \phi(\mathbf{x}_i) + b \geq +1, y_i = +1, \\ H_{-1} : \mathbf{w}^T \phi(\mathbf{x}_i) + b \leq -1, y_i = -1. \end{cases} \quad (2.2)$$

Pode-se perceber que a Eq. (2.2) pode ser recombinação e reescrita como:

$$y_i [\mathbf{w}^T \phi(\mathbf{x}_i) + b] \geq 1, i = 1, \dots, N. \quad (2.3)$$

Para uma melhor compreensão, considere um ponto $P(x_a, y_a)$ e uma reta r definida por $ax_a + by_a + c = 0$. Sabe-se que a distância do ponto P à reta r é definida por:

$$d(r, P) = \frac{|ax_a + by_a + c|}{\sqrt{a^2 + b^2}}. \quad (2.4)$$

De maneira análoga a Eq. (2.4), a distância entre um vetor $\mathbf{x}_i$ e o hiperplano H_0 definido na Eq. (2.1) pode ser encontrada por

$$\begin{aligned} d_i(\mathbf{w}^T, b; \mathbf{x}_i) &= \frac{|\mathbf{w}^T \phi(\mathbf{x}_i) + b|}{\|\mathbf{w}\|} \\ &= \frac{y_i [\mathbf{w}^T \phi(\mathbf{x}_i) + b]}{\|\mathbf{w}\|}, \end{aligned} \quad (2.5)$$

em que $\mathbf{w}$ é o vetor peso normal ao hiperplano descrito na Eq. (2.1) e $\|\mathbf{w}\|$ é a norma euclidiana do vetor peso $\mathbf{w}$ (HAYKIN, 2001).

Substituindo a Eq. (2.3) na Eq. (2.5), tem-se que:

$$d_i(\mathbf{w}^T, b; \mathbf{x}_i) = \frac{1}{k_{\mathbf{w}} k}. \quad (2.6)$$

A Figura 2 ilustra a interpretação geométrica da distância de um vetor ao hiperplano ótimo, conforme demonstração matemática exposta acima, bem como apresenta a margem de separação, conceito a ser definido posteriormente.

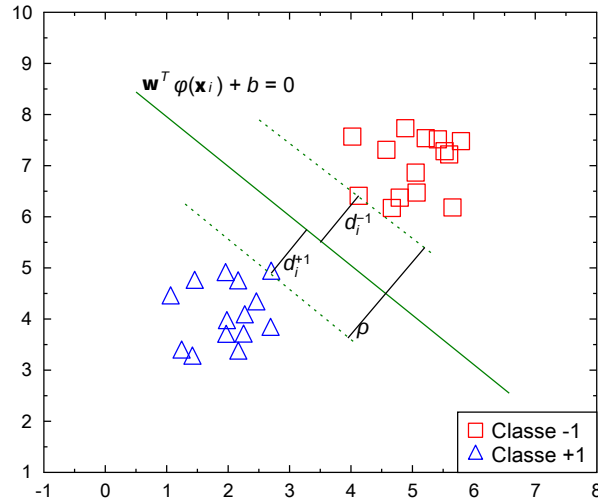


Figura 2: Interpretação geométrica da distância de um vetor $\mathbf{x}_i \in \{\pm 1\}$ ao hiperplano ótimo $H_o^{(s)}$ e da margem de separação ρ .

A partir da Eq. (2.2), nota-se que o hiperplano ótimo de separação, corresponde àquele que encontra-se equidistante entre as classes do problema em relação a vetores base denominados vetores-suporte. Tais vetores são caracterizados pelos padrões que além de satisfazerem a Eq. (2.2), contribuem diretamente para a solução (BONESSO, 2013; MOLINA, 2014). Portanto, matematicamente, o hiperplano ótimo de separação pode ser representado pela seguinte expressão:

$$H_o^{(s)} : \mathbf{w}^T \phi(\mathbf{x}_i) + b = 0. \quad (2.7)$$

Seja d_i^{+1} a menor distância entre um padrão $\mathbf{x}_i$ tal que $y_i = +1$ e o hiperplano $H_o^{(s)}$ e d_i^{-1} a menor distância entre um padrão $\mathbf{x}_i$ tal que $y_i = -1$ e o hiperplano $H_o^{(s)}$, define-se por margem de separação ρ , a maior distância geométrica entre os hiperplanos existentes. A margem de separação consiste em encontrar o hiperplano ótimo de separação dos dados entre as classes $\{\pm 1\}$:

$$\rho = d_i^{+1} + d_i^{-1}, \quad (2.8)$$

o que implica em um hiperplano equidistante entre as classes do problema. Com isso $d_i^{+1} = d_i^{-1}$. Diante do exposto, a margem de separação apresentada na Eq. (2.8) resulta na seguinte expressão:

$$\rho = \frac{2}{\| \mathbf{w} \|_k}. \quad (2.9)$$

Como mostrado em Rocha Neto (2011), a partir da incorporação do princípio SRM, visto que em sua formulação o classificador LSSVM utiliza a norma do vetor peso, é possível notar na Eq. (2.9) que ao minimizar a norma $\| \mathbf{w} \|_k$, obtém-se a minimização da dimensão VC (*Vapnik-Chervonenkis*), o que implica na diminuição da complexidade das funções de decisão. Sabendo ainda que a partir de Vapnik (1982, 1995) pode-se concluir que a maximização da margem de separação ρ é inversamente proporcional ao valor obtido da dimensão VC, é possível inferir que a formulação do problema a ser otimizado, com o intuito de obter o hiperplano ótimo, parte da maximização da Eq. (2.9):

$$\max \rho = \max \frac{2}{\| \mathbf{w} \|_k}. \quad (2.10)$$

A maximização na Eq. (2.10) equivale a minimização da norma $\| \mathbf{w} \|_k$. Esta norma por sua vez pode ser substituída por $\sqrt{\mathbf{w}^T \mathbf{w}}$ ou ainda $\frac{1}{2} \mathbf{w}^T \mathbf{w}$. Assim, a expressão a ser minimizada pode ser representada pela seguinte função custo:

$$J_P(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}. \quad (2.11)$$

A Eq. (2.11) é independente do viés b desde que a Eq. (2.3) seja satisfeita. Modificando o valor do viés b , a margem de separação ρ não será alterada. O que ocorre é o deslocamento do hiperplano na sua direção normal, logo este não será mais ótimo, uma vez que estará mais próximo de uma classe do que da outra. Minimizar a Eq. (2.11) equivale a implementar o princípio da Minimização do Risco Estrutural, conforme exposto em Gunn (1998).

Todos os conceitos apresentados até este ponto são válidos tanto para classificadores SVM quanto para classificadores LSSVM. Na Seção 2.3 são apresentados os detalhes da teoria para os classificadores SVM, enquanto os detalhes da formulação para os classificadores LSSVM são descritos na Seção 2.4.

2.3 Fundamentos Teóricos do Classificador SVM

2.3.1 Variáveis de Folga

As Máquinas de Vetores-Suporte caracterizam-se por possuírem os hiperplanos de apoio localizados de tal maneira que tenham a menor distância entre um padrão de treinamento de uma determinada classe e o hiperplano ótimo. O hiperplano ótimo é aquele equidistante entre as classes do problema quando se leva em consideração os vetores-suporte.

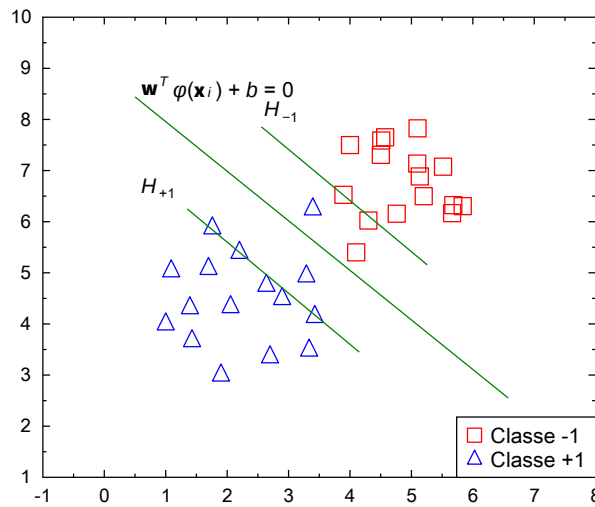


Figura 3: Divisão do espaço de características do classificador SVM.

Como mencionado, maximizar a margem de separação ρ , consiste em minimizar a função custo apresentada na Eq. (2.11). Portanto, encontrar um hiperplano ótimo que separe os dados de entrada, consiste em minimizar a seguinte equação:

$$\min J_P(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}, \quad (2.12)$$

$$\text{sujeito a } y_i [\mathbf{w}^T \phi(\mathbf{x}_i) + b] \geq 1, \quad i = 1, \dots, N.$$

Nem sempre os padrões pertencentes a uma determinada classe podem ser completamente separados por um hiperplano, principalmente em problemas onde é verificado sobreposição entre os dados. Isso justifica-se tanto pela presença de ruídos ou pela própria natureza do problema. Para que a complexidade da função no espaço de entrada não seja muito elevada e também com o intuito de evitar sobreajuste da superfície de decisão (*overfitting*) aos dados de treinamento, permite-se que alguns dados violem a restrição da Eq. (2.3). Com esse objetivo, foram inseridas na formulação do classificador SVM variáveis de relaxamento que flexibilizam as restrições de otimização, chamadas de *variáveis de folga*.

Assim, sua formulação é caracterizada por um relaxamento das restrições, ou seja

$$y_i[\mathbf{w}^T \phi(\mathbf{x}_i) + b] \geq 1 - \xi_i, \quad i = 1, \dots, N, \quad (2.13)$$

em que $\xi_i \geq 0, i = 1, \dots, N$.

As variáveis de folga são obtidas automaticamente durante o processo de treinamento do classificador SVM (CORTES; VAPNIK, 1995) e assumem valores positivos, de tal forma que, cada padrão de treinamento $\mathbf{x}_i$ possui uma respectiva variável $\xi_i \geq 0$ associada. A Figura 4 ilustra a interpretação geométrica das variáveis de folga em classificadores SVM.

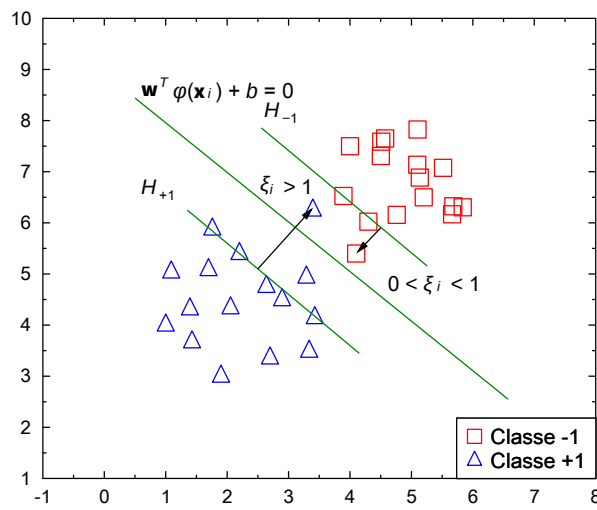


Figura 4: Interpretação geométrica das variáveis de folga em classificadores SVM.

Padrões de treinamento que ultrapassam a sua respectiva margem de apoio, mas encontram-se do mesmo lado da sua classe de origem em relação ao hiperplano ótimo de separação, possuem valores dentro do intervalo $0 < \xi_i < 1$. Já padrões que estão localizados na região oposta à sua classe de origem e também em relação ao hiperplano ótimo, possuem valores $\xi_i > 1$ e padrões que não são considerados vetores-suporte têm valor igual a zero.

2.3.2 Formulação Primal-Dual

A formulação do problema de otimização das Máquinas de Vetores-Suporte é chamada de *problema primal*, também denominado na literatura como *problema primordial* (HAYKIN, 2001). Já considerando os conceitos apresentados na Seção 2.3.1, a resolução do *problema primal* do classificador SVM consiste na obtenção do vetor peso $\mathbf{w}$ e do vies b pela minimização

da seguinte função custo:

$$\begin{aligned} \min J_P(\mathbf{w}, b, \boldsymbol{\xi}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i, \\ \text{sujeito a } \begin{cases} y_i [\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b] \geq 1 - \xi_i, & i = 1, \dots, N, \\ \xi_i \geq 0, & i = 1, \dots, N. \end{cases} \end{aligned} \quad (2.14)$$

A constante C representa uma regularização entre os dois termos da função $J_P(\mathbf{w}, b, \boldsymbol{\xi})$, sendo o primeiro termo ($\frac{1}{2} \mathbf{w}^T \mathbf{w}$) responsável pela maximização da margem e o segundo termo ($C \sum_{i=1}^N \xi_i$) responsável por minimizar os erros permitidos.

Problemas dessa natureza que apresentam uma função custo convexa e com restrições lineares (HAYKIN, 2001), podem ser solucionados através da aplicação do *método dos multiplicadores de Lagrange*. Este método agrega as restrições à função custo, que associadas a variáveis auxiliares, transformam o *problema primal* em um novo. Dessa forma, a seguinte função Lagrangeana é obtida:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{y_i [\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b] - 1 + \xi_i\} - \sum_{i=1}^N \beta_i \xi_i, \quad (2.15)$$

em que os conjuntos de multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^N$ e $\{\beta_i\}_{i=1}^N$ possuem valores positivos.

A expansão da Eq. (2.15) resulta em

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i y_i \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i \xi_i - \sum_{i=1}^N \beta_i \xi_i. \quad (2.16)$$

Para a solução do problema acima é necessário que a função custo seja minimizada em relação a $\mathbf{w}$, b e $\boldsymbol{\xi}$, maximizada em relação a $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$, e igualada a zero. Dessa forma as seguintes condições de otimalidade são obtidas:

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{w}} &= 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \boldsymbol{\varphi}(\mathbf{x}_i), \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} &= 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0, \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} &= 0 \Rightarrow C = \alpha_i + \beta_i. \end{aligned} \quad (2.17)$$

Vale notar que os termos $-\sum_{i=1}^N \alpha_i \xi_i$ e $-\sum_{i=1}^N \beta_i \xi_i$ da Eq. (2.16), podem ser re combinados,

o que resulta em $-\sum_{i=1}^N \xi_i(\alpha_i + \beta_i)$. Dessa forma a Eq. (2.16) apresenta-se como

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i y_i \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \xi_i(\alpha_i + \beta_i). \quad (2.18)$$

Aplicando o resultado $C = \alpha_i + \beta_i$ obtido das condições de otimização no termo $C \sum_{i=1}^N \xi_i$, é possível verificar que o segundo e último termos da Eq. (2.18) podem ser eliminados. Levando ainda em consideração que o quarto termo da Eq. (2.18) é zero, em função da condição de otimização $\sum_{i=1}^N \alpha_i y_i = 0$, a forma expandida da função Lagrangeana pode ser reescrita como

$$L(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + \sum_{i=1}^N \alpha_i. \quad (2.19)$$

Substituindo o resultado $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \boldsymbol{\varphi}(\mathbf{x}_i)$ no segundo termo da Eq. (2.19), obtém-se $-\mathbf{w}^T \mathbf{w}$, o que resulta em

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\alpha}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \mathbf{w}^T \mathbf{w} + \sum_{i=1}^N \alpha_i, \\ &= -\frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{i=1}^N \alpha_i. \end{aligned} \quad (2.20)$$

Por fim, substituindo os valores de $\mathbf{w}$ da Eq. (2.20) pelo resultado $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \boldsymbol{\varphi}(\mathbf{x}_i)$ outrora obtido e reorganizando seus termos, obtém-se a seguinte função custo dual a saber

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x}_j). \quad (2.21)$$

Com isso, o *problema dual* a ser otimizado para o classificador SVM é expresso por

$$\begin{aligned} \max L(\boldsymbol{\alpha}) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x}_j), \\ \text{sujeito a } &\left(\begin{aligned} \sum_{i=1}^N \alpha_i y_i &= 0, \\ 0 \leq \alpha_i &\leq C, \quad i = 1, \dots, N. \end{aligned} \right. \end{aligned} \quad (2.22)$$

Um método utilizado com o objetivo de evitar o uso do espaço de características de alta dimensão, permitindo o uso de uma função que satisfaz a condição de *Mercer* (MERCER, 1909), é denominado de *truque de kernel*. O teorema de *Mercer* permite identificar quando uma função qualquer candidata a kernel representa o produto interno em algum espaço de característica. Diante disso, a utilização de um kernel evita que seja necessário a realização

direta do mapeamento $\varphi(\cdot)$. Exemplos de funções de kernel tipicamente utilizadas na literatura são mostradas na Tabela 1.

Tabela 1: Exemplos de *kernels* tipicamente utilizados.

Tipo de Kernel	Função $K(\mathbf{x}, \mathbf{x}_i)$, $i, j = 1, \dots, N$	Comentários
Linear	$K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}^T \mathbf{x}_i$	-
Polinomial	$K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x}^T \mathbf{x}_i)^p$	O termo p é especificado pelo usuário
RBF	$K(\mathbf{x}, \mathbf{x}_i) = \exp -k \ \mathbf{x} - \mathbf{x}_i\ ^2 / \sigma^2$	O termo σ é especificado pelo usuário
MLP	$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1)$	Nem todos os valores de β_0 e β_1 satisfazem o teorema de Mercer

Portanto, aplicando o *truque de kernel* no problema dual apresentado na Eq. (2.22), têm-se

$$\begin{aligned}
 \max L(\boldsymbol{\alpha}) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j), \\
 &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j).
 \end{aligned} \quad (2.23)$$

Neste sentido, a Eq. (2.22) pode ser redefinida para

$$\begin{aligned}
 \max L(\boldsymbol{\alpha}) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \\
 \text{sujeito a } & \left(\begin{aligned} \sum_{i=1}^N \alpha_i y_i &= 0, \\ 0 \leq \alpha_i &\leq C, \quad i = 1, \dots, N. \end{aligned} \right.
 \end{aligned} \quad (2.24)$$

Ao resolver o problema apresentado na Eq. (2.24), são encontrados os valores ótimos dos multiplicadores de Lagrange $\{\alpha_i^o\}_{i=1}^N$. Estes por sua vez, são utilizados para encontrar os valores das variáveis primais $\mathbf{w}_o$ e b_o por meio das seguintes relações:

$$\left(\begin{aligned} \mathbf{w}_o &= \sum_{i=1}^N \alpha_i^o y_i \varphi(\mathbf{x}_i), \\ b_o &= \frac{1}{2} \mathbf{w}_o^T \mathbf{x}^+ + \mathbf{w}_o^T \mathbf{x}^- \end{aligned} \right. \quad (2.25)$$

onde o padrão $\mathbf{x}^+ \in \{+1\}$ e o padrão $\mathbf{x}^- \in \{-1\}$.

Diante do exposto, a classificação de um determinado padrão $\mathbf{x}$ pode ser determinada pela

função sinal a saber:

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i^o y_i K(\mathbf{x}, \mathbf{x}_i) + b_o \right), \quad (2.26)$$

em que valores positivos são associados a padrões pertencentes à classe $y_i = +1$, enquanto valores negativos são associados a padrões pertencentes à classe $y = -1$.

2.4 Fundamentos Teóricos do Classificador LSSVM

2.4.1 Variáveis de Folga

O espaço de características dos classificadores LSSVM é composto por dois hiperplanos paralelos situados cada um sobre suas respectivas classes (veja Figura 5). Dessa forma, a correta classificação para um padrão $\mathbf{x}_i$ tal que $y_i = +1$ ocorre quanto menor for sua distância ξ_i em relação ao hiperplano H_{+1} . Analogamente, a correta classificação para um padrão $\mathbf{x}_i$, tal que $y_i = -1$, ocorre quanto menor for sua distância ξ_i em relação ao hiperplano H_{-1} .

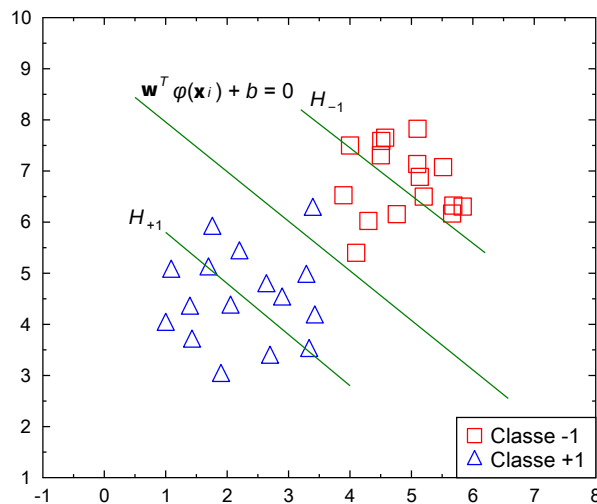


Figura 5: Divisão do espaço de características do classificador LSSVM.

A partir da Eq. (2.1), o espaço de características do classificador LSSVM passa a ser dividido em duas regiões definidas por:

$$\begin{aligned} H_{+1} : \mathbf{w}^T \phi(\mathbf{x}_i) + b &= +1, y_i = +1, \\ H_{-1} : \mathbf{w}^T \phi(\mathbf{x}_i) + b &= -1, y_i = -1, \end{aligned} \quad (2.27)$$

que recombinação, pode ser reescrita como:

$$y_i [\mathbf{w}^T \phi(\mathbf{x}_i) + b] = 1, i = 1, \dots, N \quad (2.28)$$

De maneira análoga ao verificado na formulação dos classificadores SVM, o problema que permite encontrar um hiperplano que separe adequadamente os dados ilustrados na Figura 5 pode ser expresso por

$$\min J_P(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}, \quad (2.29)$$

$$\text{sujeito a } y_i [\mathbf{w}^T \varphi(\mathbf{x}_i)] + b = 1, \quad i = 1, \dots, N.$$

E assim, com a inclusão da variável de folga ξ , a Eq. (2.28) passa a ser expressa pela seguinte restrição de igualdade:

$$y_i [\mathbf{w}^T \varphi(\mathbf{x}_i)] + b = 1 - \xi_i, \quad i = 1, \dots, N \quad (2.30)$$

Diferentemente do conceito empregado nas Máquinas de Vetores-Suporte, a variável de folga ξ dos classificadores LSSVM pode assumir qualquer valor real e representa a distância entre um padrão $\mathbf{x}_i$ e seu respectivo hiperplano. Pode-se então, através da variável de folga ξ , determinar se um padrão está próximo ou distante do hiperplano correspondente à sua classe. Para uma visão mais detalhada sobre as variáveis de folga, considere a Figura 6 derivada de outra apresentada por Carvalho (2005).

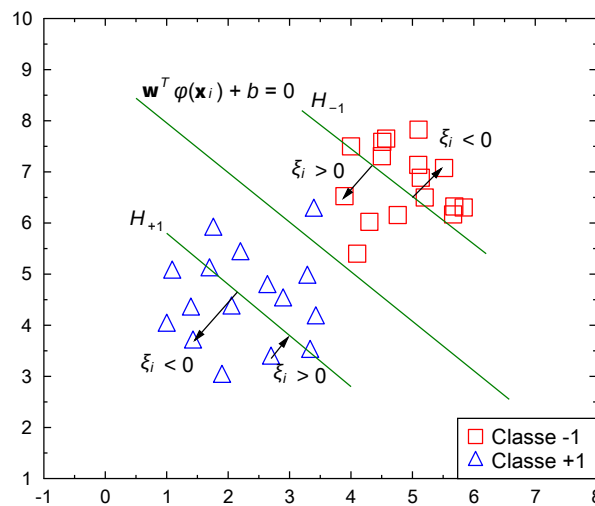


Figura 6: Análise das variáveis de folga utilizadas no classificador LSSVM.

Valores negativos para as variáveis de folga ($\xi_i < 0$) indicam que o padrão $\mathbf{x}_i$ encontra-se no lado oposto ao hiperplano correspondente à sua classe e à região da outra classe. Tal situação representa a correta classificação do padrão, pois quanto menor o valor da variável de folga ξ_i , mais distante dos hiperplanos H_{+1} e H_{-1} se encontra o padrão $\mathbf{x}_i$.

Valores iguais a zero para as variáveis de folga ($\xi_i = 0$) indicam que o padrão $\mathbf{x}_i$ encontra-se

exatamente sobre o hiperplano relacionado a sua classe. O que de certa forma também caracteriza a correta classificação do padrão levando em consideração a distância entre os hiperplanos paralelos H_{+1} e H_{-1} . Entretanto, esse é um caso raro de ocorrer.

Já valores positivos para as variáveis de folga ($\xi_i > 0$) indicam que o padrão $\mathbf{x}_i$ encontra-se no mesmo lado da região da outra classe, levando em consideração o hiperplano relacionado a sua classe. Nesta situação a determinação clara sobre a classificação do padrão somente pela análise do valor da variável de folga ξ_i é imprecisa. A única afirmação que pode ser feita é que quanto maior o valor da variável de folga ξ_i , mais próximo do hiperplano incorreto o padrão se encontra.

2.4.2 Formulação Primal

Assim como nos classificadores SVM, a formulação do problema de otimização das Máquinas de Vetores-Suporte por Mínimos Quadrados parte de um *problema primal*, sendo o mesmo dado por

$$\min J_P(\mathbf{w}, b, \xi_i) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\nu}{2} \sum_{i=1}^N \xi_i^2, \quad (2.31)$$

$$\text{sujeito a } y_i[\mathbf{w}^T \phi(\mathbf{x}_i) + b] = 1 - \xi_i, \quad i = 1, \dots, N.$$

As duas modificações realizadas para concepção do classificador LSSVM, a partir do modelo SVM, estão presentes na formulação primal de otimização. A primeira modificação realizada consiste na utilização de restrições de igualdade ao invés de desigualdade, onde o valor 1 (lado direito da equação) passa a ser considerado um valor alvo e não mais um limiar. Juntamente com esse valor alvo, a variável ξ_i representa os erros de classificação admissíveis em caso de distribuições com sobreposição². A segunda mudança é feita na função custo, a partir da incorporação da soma das variáveis de folga ao quadrado, ponderadas por um termo de regularização³ ν .

Ao continuar o processo de resolução do problema pelo *método dos multiplicadores de Lagrange*, a seguinte função Lagrangeana pode ser obtida:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\nu}{2} \sum_{i=1}^N \xi_i^2 - \sum_{i=1}^N \alpha_i \{y_i[\mathbf{w}^T \phi(\mathbf{x}_i) + b] - 1 + \xi_i\}, \quad (2.32)$$

² Essas variáveis de erro desempenham um papel semelhante as variáveis de folga da formulação do modelo SVM.

³ Este parâmetro tem o mesmo significado da constante de regularização C utilizada na formulação do modelo SVM.

em que $\{\alpha_i\}_{i=1}^N \in \mathbb{R}$ são os multiplicadores de Lagrange.

A partir da Eq. (2.32) pode-se obter as seguintes condições de otimalidade para a minimização em relação a $\mathbf{w}$, b e ξ , a saber:

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \xi, \alpha)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \varphi(\mathbf{x}_i), \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha)}{\partial b} &= - \sum_{i=1}^N \alpha_i y_i, \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha)}{\partial \xi} &= \gamma \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i. \end{aligned} \quad (2.33)$$

E para a maximização em relação a α :

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha)}{\partial \alpha} = - y_i [\mathbf{w}^T \varphi(\mathbf{x}_i) + b] + 1 - \xi_i. \quad (2.34)$$

Dessa forma, a solução da função lagrangeana para o *problema primal* de otimização se dá pela resolução das derivadas parciais apresentadas nas Eq. (2.33) e (2.34) em relação aos parâmetros primais, de tal forma que para encontrar o ponto de sela da função Lagrangeana, essas equações são igualadas a zero, o que resulta em um sistema *Karush-Kuhn-Tucker* (sistema KKT) a saber:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \varphi(\mathbf{x}_i), \\ \sum_{i=1}^N \alpha_i y_i = 0, \\ y_i [\mathbf{w}^T \varphi(\mathbf{x}_i) + b] - 1 + \xi_i = 0, \forall_i, \\ \alpha_i = \gamma \xi_i, \forall_i. \end{cases} \quad (2.35)$$

Suykens *et al.* (2002) notam que o tamanho do sistema KKT não é influenciado pela dimensão do espaço de entrada $\mathbb{R}^n$, pois este depende apenas do tamanho de N . Assim, para vetores de entrada de grandes dimensões, a solução é beneficiada pela resolução da função lagrangeana ao invés do *problema primal*.

A igualdade $\alpha_i = \gamma \xi_i$ apresentada na Eq. (2.35) sugere que os multiplicadores de Lagrange geralmente são não-nulos, podendo ser positivos ou negativos devido as restrições de igualdade impostas (SUYKENS; VANDEWALLE, 1999a). Isso ocorre devido a relação diretamente proporcional entre os multiplicadores de Lagrange e as variáveis de folga. Conforme os conceitos apresentados a partir da análise da Figura 6, raramente a situação $\xi_i = 0$ ocorre, consequentemente, raramente existem α_i com valores nulos.

Tal característica indica que todos os dados de treinamento contribuem para a solução do

problema e são assim denominados vetores-suporte. A única situação que resulta em $\alpha_i = 0$ seria se o vetor de treinamento em questão estivesse localizado exatamente sobre o hiperplano que representa a sua classe, situação bastante incomum.

Em uma situação que se obtém diversos valores não-nulos para os multiplicadores de Lagrange se verifica a falta de esparsidade no vetor de multiplicadores de Lagrange do classificador LSSVM. Essa falta de esparsidade tem um impacto negativo na resolução de problemas cuja base de dados é demasiadamente grande, pois deve-se armazenar todos os elementos do conjunto de treinamento juntamente com os multiplicadores de Lagrange associados para a posterior resolução do problema.

A fim de se obter a superfície de decisão, os termos $\mathbf{w}$ e $\varphi(\mathbf{x}_i)$ não são calculados diretamente (SUYKENS; VANDEWALLE, 1999a). Após a eliminação dos termos $\mathbf{w}$ e ξ_i , conforme exposto por Suykens *et al.* (2002), o sistema KKT pode ser resolvido diretamente por um sistema linear, a saber:

$$\mathbf{D}\mathbf{z} = \mathbf{l}, \quad (2.36)$$

tal que:

$$\mathbf{D} = \begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \Omega + \gamma^{-1}\mathbf{I} \end{array}, \quad \mathbf{z} = \begin{array}{c} b \\ \boldsymbol{\alpha} \end{array}, \quad \mathbf{l} = \begin{array}{c} 0 \\ \mathbf{1} \end{array}, \quad (2.37)$$

em que $\Omega \in \mathbb{R}^{N \times N}$ é uma matriz cujas entradas são dadas por $\Omega_{i,j} = y_i y_j \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$, $i, j = 1, \dots, N$. Além de $\mathbf{y} = [y_1 \dots y_N]^T$ e o símbolo $\mathbf{1}$ que representa um vetor de 1's com dimensão N .

Vale ressaltar que, na formulação original proposta por Suykens e Vandewalle (1999a), o vetor $\mathbf{y}^T$ da Eq. (2.37) é negativo. Contudo, a substituição do vetor $-\mathbf{y}^T$ pelo vetor $\mathbf{y}^T$, não influencia na resolução final do sistema (CARVALHO, 2005).

Aplicando o *truque de kernel* em $\Omega_{i,j}$, tem-se:

$$\begin{aligned} \Omega_{i,j} &= y_i y_j \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j) \\ &= y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1, \dots, N \end{aligned} \quad (2.38)$$

A partir da igualdade $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \varphi(\mathbf{x}_i)$ presente na Eq. (2.35) verifica-se que o vetor peso $\mathbf{w}$ pode ser escrito como combinações lineares entre os multiplicadores de Lagrange α_i e os padrões de treinamento $\mathbf{x}_i$ (MARETTO, 2007). Diante do exposto, substituindo do termo $\mathbf{w}$

na equação do hiperplano H_o , representada na Eq. 2.1, tem-se

$$\begin{aligned} y_i &= \sum_{i=1}^N \alpha_i y_i \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_i) + b, \\ &= \sum_{i=1}^N \alpha_i y_i h_{\varphi(\mathbf{x}_i)^T, \varphi(\mathbf{x}_i)} + b, \end{aligned} \quad (2.39)$$

em que $h_{\varphi(\mathbf{x}_i)^T, \varphi(\mathbf{x}_i)}$ representa o produto interno $\mathbf{x}_i$ e $\mathbf{x}$.

Logo, substituindo o produto interno por uma função de kernel $K(\mathbf{x}, \mathbf{x}_i)$, a saída do classificador LSSVM caracterizada pela função lagrangeana expressa na Eq. (2.32) também pode ser encontrada por

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \right), \quad (2.40)$$

tal que, assim como verificado nos classificadores SVM, valores positivos são associados a padrões pertencentes à classe $y_i = +1$, enquanto valores negativos são associados a padrões pertencentes à classe $y = -1$.

O problema da falta de esparsidade em classificadores LSSVM conforme mencionado anteriormente e verificado ao se observar a Eq. (2.35), motivou o estudo sobre a imposição de esparsidade aos multiplicadores de Lagrange. Este assunto foi e ainda é abordado em diversos trabalhos que tem como objetivo, reduzir o número de vetores-suporte utilizados pelo modelo sem afetar seu desempenho. Tais métodos serão discutidos em detalhes no Capítulo 3.

2.5 Obtenção dos Parâmetros Ótimos

2.5.1 Matriz Inversa

A solução do sistema linear apresentado na Eq. (2.36), a fim de se obter os termos b e α , pode ser obtida diretamente pela inversão da matriz $\mathbf{D}$, desde que essa matriz seja não singular⁴. Portanto da Eq. (2.36) pode-se obter:

$$\begin{aligned} \mathbf{D}^{-1} \mathbf{D} \mathbf{z} &= \mathbf{D}^{-1} \mathbf{l}, \\ (\mathbf{D}^{-1} \mathbf{D}) \mathbf{z} &= \mathbf{D}^{-1} \mathbf{l}, \\ \mathbf{I} \mathbf{z} &= \mathbf{D}^{-1} \mathbf{l}. \end{aligned} \quad (2.41)$$

⁴ Para casos em que a matriz é singular utilizam-se técnicas de regularização de matrizes (HADAMARD, 1915; NOSCHESSE; REICHEL, 2012)

Ou seja, tem-se que

$$\mathbf{z} = \mathbf{D}^{-1} \mathbf{l} = \begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \Omega + \lambda^{-1} \mathbf{I} \end{array} \begin{array}{c} \#^{-1} \\ 0 \\ 1 \end{array}. \quad (2.42)$$

2.5.2 Matriz Pseudo-Inversa

Considerando que a solução do sistema parte de uma matriz $\mathbf{D}$ não quadrada, faz-se necessário a utilização do método da Pseudo-Inversa (PENROSE, 1955). Tal consideração é feita, pois de acordo com o objetivo do trabalho, somente as colunas que representam valores de multiplicadores de Lagrange nulos serão removidas com o intuito de reduzir a complexidade do classificador.

Portanto, a solução da Eq. (2.36) para uma matriz $\mathbf{D}$ não quadrada é dada por:

$$\begin{aligned} \mathbf{D}^T \mathbf{D} \mathbf{z} &= \mathbf{D}^T \mathbf{l}, \\ (\mathbf{D}^T \mathbf{D}) \mathbf{z} &= \mathbf{D}^T \mathbf{l}, \\ (\mathbf{D}^T \mathbf{D})^{-1} (\mathbf{D}^T \mathbf{D}) \mathbf{z} &= (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{l}, \\ \mathbf{I} \mathbf{z} &= (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{l}. \end{aligned} \quad (2.43)$$

Logo, tem-se que

$$\mathbf{z} = \mathbf{D}^\dagger \mathbf{l} \quad (2.44)$$

onde $\mathbf{D}^\dagger = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T$ é a Pseudo-Inversa.

2.5.3 Otimização dos Hiperparâmetros

O termo de regularização λ é responsável pela minimização da complexidade do modelo, uma vez que ao aumentá-lo, mais ênfase é dada na redução da magnitude do erro quadrático de treinamento, estabelecendo assim um equilíbrio entre a complexidade e o erro de treinamento.

Assim como λ , dependendo do tipo de kernel utilizado, outros parâmetros necessitam que seu valor sejam definidos *a priori*. Existem várias técnicas diferentes com a finalidade de refinar os hiperparâmetros relacionados a constante de regularização e parâmetros da função de kernel. Um método bastante comum consiste na utilização de uma busca em grade (*grid search*) combinado com um método de validação cruzada (*cross validation*) (HSU; CHANG; LIN, 2003).

A busca em grade está associada a uma busca exaustiva associada a um subconjunto de hiperparâmetros que se deseja otimizar. Supondo o uso de um kernel RBF, além do parâmetro γ , o parâmetro σ também deve ser estimado *a priori*, logo a busca em grade será composta por um subconjunto de diferentes valores para o par (γ, σ) , cada um com um certo intervalo, a fim de se obter um par ótimo tal que o classificador identifique com precisão dados não utilizados no processo de treinamento.

O procedimento de validação cruzada é caracterizado primeiramente pela divisão da base em dados de treinamento e dados de teste, sendo este último mantido para uma avaliação final. O conjunto de treinamento deve ainda ser dividido em v subconjuntos de mesmo tamanho. Sequencialmente, cada subconjunto v é utilizado para validação do classificador que será treinado com os $v - 1$ subconjuntos restantes. Esse procedimento é repetido até que cada um dos v subconjuntos sejam utilizados na validação do modelo. Dessa forma, vários valores de pares (γ, σ) são avaliados e aquele que apresentar um melhor desempenho médio é escolhido.

Conforme mostrado por Hsu, Chang e Lin (2003), dependendo do intervalo estabelecido para os valores dos pares (γ, σ) , o tempo para percorrer toda a busca em grade pode ser muito dispendioso. Dessa forma, utiliza-se primeiramente uma grade de malha grossa⁵. Uma vez identificada a melhor região, uma grade mais fina⁶ nessa região é utilizada.

Após encontrar o melhor valor para o par (γ, σ) , todo o conjunto de treinamento é retreinado agora com os valores ótimos do par (γ, σ) para que o classificador final seja encontrado. Conforme Suykens *et al.* (2002) e ainda Gestel *et al.* (2004), os hiperparâmetros também podem ser obtidos através de outras técnicas tais como: inferência Bayesiana, técnicas de *bootstrap*, dentre outras.

2.6 Conclusão

O classificador LSSVM é uma abordagem bastante atrativa em termos computacionais para a resolução de problemas de classificação de padrões. O uso de restrições de igualdade em conjunto com a incorporação de variáveis de folga ao quadrado contribuem para que a resolução de um sistema de equações lineares ao invés de problemas de programação quadrática, fizessem desse modelo uma alternativa ao uso de classificadores SVM.

Além das vantagens do modelo e do grande potencial quanto a capacidade de generalização

⁵ Define-se por grade de malha grossa o uso de intervalos com alta granularidade para os valores dos pares (γ, σ) .

⁶ Define-se por grade de malha fina o uso de intervalos com baixa granularidade dentro de uma região específica para os valores dos pares (γ, σ) .

pela maximização da margem de separação ρ , mostrou-se as limitações características da LSSVM. A obtenção dos parâmetros ótimos se dá em função das características do sistema linear, onde para os casos que verifica-se uma matriz $\mathbf{D}$ não quadrada, o método da Pseudo-Inversa vêm sendo aplicado com sucesso na solução do problema. Já no que diz respeito a otimização dos hiperparâmetros que dependem do tipo de função de kernel utilizada e do termo de regularização γ do classificador, mostrou-se como alternativa a técnica de busca em grade combinada com um método de validação cruzada, técnica bastante utilizada neste cenário.

A fim de atenuar a falta de esparsidade do vetor de multiplicadores de Lagrange, propõe-se no Capítulo 4, novas abordagens de poda pelo uso de técnicas de computação evolutiva.

ALGORITMOS GENÉTICOS

Algoritmos Genéticos são uma classe particular de algoritmos evolutivos¹ que se inspiram na evolução natural e que aplicam conceitos de herança, mutação, seleção natural e reprodução. AGs podem ser utilizados para gerar soluções úteis para problemas de otimização. O objetivo deste capítulo é descrever a fundamentação teórica necessária sobre AGs com o intuito de facilitar o entendimento dos métodos propostos para obtenção de conjuntos reduzidos de vetores-suporte em classificadores LSSVM.

Neste capítulo é descrito como ocorre o processo de evolução natural dos indivíduos ao longo das gerações, o funcionamento de operadores clássicos de seleção, reprodução e mutação, bem como de operadores de reprodução mais específicos no contexto de Algoritmos Genéticos. Vale ressaltar que grande parte da fundamentação teórica presente neste capítulo encontra-se presente em Goldberg (1989).

3.1 Introdução

Em meados do século *XIX*, contrariando a corrente de pensamento *criacionista*, Charles Darwin propôs uma ideia que defende a evolução das espécies, também conhecida por *darwinismo* ou *evolucionismo*.

Como preceito básico do *evolucionismo*, indivíduos independente de espécie, disputam recursos limitados em um determinado meio. Neste contexto, no processo de evolução das espécies, indivíduos cujas características particulares a eles proporcionam uma melhor chance de sobrevivência nesse ambiente são considerados mais adaptados e possuem mais chances de sobreviverem e se reproduzirem.

Consequentemente, no processo de reprodução, grande parte das características dos pais são repassadas aos filhos o que proporciona a propagação de características mais desejáveis. Contudo, a evolução não é um processo isento a falhas (GARCIA, 2014). Um fenômeno

¹ Família de procedimentos iterativos de busca ou otimização (IYODA, 2000).

conhecido por mutação, é responsável pela incorporação de características ao novo indivíduo durante a replicação e transmissão dos genes, o que do ponto de vista de evolução natural, faz com que o indivíduo possua uma maior variabilidade genética (BARCELLOS, 2000).

Pesquisas relacionadas a AGs foram iniciadas ainda na década de 50 através da utilização de computadores para simulações de sistemas biológicos. Entretanto, somente na década de 70, através de um grupo de pesquisadores da universidade de Michigan, tais métodos de otimização e busca inspirados no processo de evolução das espécies foram consolidados (MALAQUIAS, 2006).

A teoria de Algoritmos Genéticos está compreendida como um dos paradigmas e métodos presentes na área de Computação Evolutiva. Neste contexto, podem-se ainda citar métodos como Programação Evolutiva (PE) (FOGEL, 1962), Programação Genética (PG) (KOZA, 1989), Estratégias Evolutivas (EEs) (SCHWEFEL, 1975), entre outros. A popularização relacionada a utilização de AGs quando comparado com outros métodos de CE, justifica-se principalmente pela eficácia na solução encontrada, como também pela flexibilidade e simplicidade de implementação (CARDOSO, 1998).

Duas importantes questões sobre AG devem ser levadas em consideração: prospecção e exploração. A primeira está diretamente relacionada a utilização do conhecimento já disponível e assim descobrir a melhor solução. Já a exploração consiste em investigar uma nova área ainda desconhecida no espaço de busca. Em problemas do mundo real, faz-se necessário manter um equilíbrio entre a prospecção e exploração (KUMAR, 2013).

A exploração no espaço de busca para problemas do mundo real, geralmente ocorre pela otimização de uma coleção de objetivos. Inicialmente, algoritmos determinísticos foram os primeiros métodos abordados pela comunidade científica para essa exploração, contudo, devido a limitações em solucionar problemas de otimização com características de descontinuidade e multimodalidade, os estudos de métodos estocásticos tornaram-se cada vez mais objeto de interesse dos pesquisadores (PARREIRAS, 2006).

Ainda segundo Parreiras (2006), esses métodos estocásticos de busca mono-objetivo e multiobjetivo inspirados pelo processo de evolução natural e pela genética, apresentam como características a capacidade de encontrar ótimos globais mesmo para funções de alta complexidade, bem como o fato de não ser necessário o cálculos de derivadas. Já para casos de busca multiobjetivo, assunto abordado em mais detalhes na Seção 3.3, dependendo do método utilizado, em uma única execução é possível obter várias soluções para o referido problema.

3.2 Algoritmos Genéticos

Algoritmos Genéticos foram propostos por Holland (1975) e fazem uso de mecanismos de busca paralela e adaptativa. O processamento típico de Algoritmos Genéticos é apresentado na Figura 7.

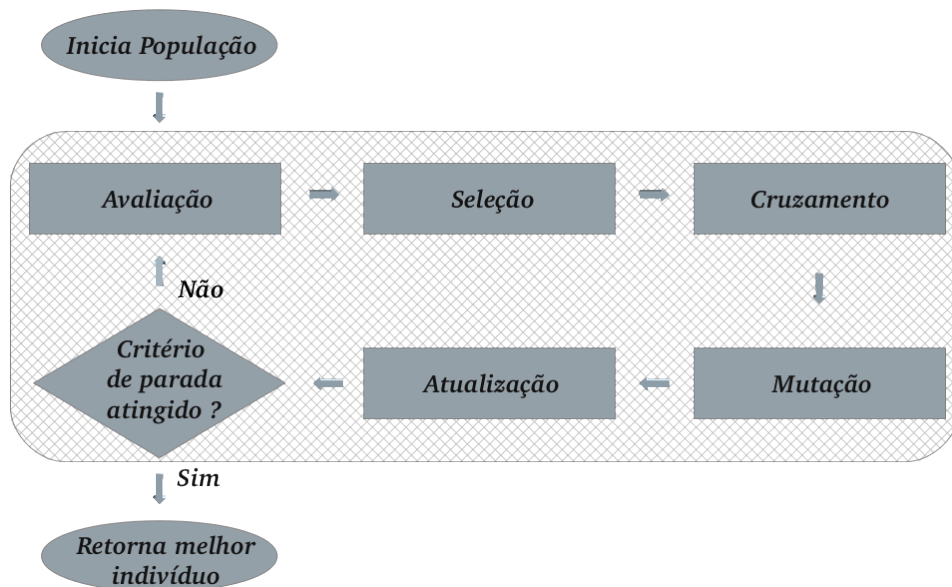


Figura 7: Diagrama básico de funcionamento de um algoritmo genético.

As diversas etapas: Inicia População, Avaliação, Seleção, Cruzamento, Mutação, Atualização e Critério de Parada são apresentadas em detalhes.

A etapa **Inicia População**, geralmente faz o uso de funções responsáveis por gerar indivíduos da primeira população de forma aleatória, com o objetivo de fornecer uma maior generalização de soluções. Dentre os operadores mais usuais para iniciar uma população pode-se elencar (GOLDBERG, 1989):

- iniciação randômica uniforme: os valores de cada gene são escolhidos de forma aleatória;
- iniciação randômica não uniforme: determinados valores são inseridos na população com maior frequência que o restante;
- iniciação randômica com “dope”: em meio a uma população já gerada aleatoriamente, são inseridos indivíduos otimizados. Neste tipo de iniciação, um ou mais super indivíduos podem dominar o processo de evolução e consequentemente, ocasionar uma convergência prematura;

- iniciação parcialmente enumerativa: considerando uma determinada ordem, os indivíduos são inseridos de tal maneira que a população inicia o processo evolutivo possuindo todos os formatos possíveis dessa ordem outrora definida.

Na etapa **Avaliação** é feita a análise da função de aptidão para cada indivíduo e a sua respectiva classificação de acordo com o grau de adaptação que cada um representa, permitindo assim a formulação de um *rank*. O valor do *rank* é usado para orientar o processo de reprodução em que os indivíduos com alto valor de adaptação têm mais chance de difundir seus genes sobre a população seguinte.

Para casos em que o problema a ser solucionado está relacionado à otimização de uma função matemática, o próprio valor de retorno da função pode ser utilizado como valor de aptidão. Para outros tipos de problemas, formula-se então uma função objetivo².

Na etapa **Seleção**, indivíduos são escolhidos mediante um critério, para serem entregues como entrada para a etapa **Cruzamento**. Esta por sua vez é responsável pela reprodução, ou seja, recombinação dessas soluções para que novos indivíduos sejam gerados. Em seguida, na etapa **Mutação**, o material genético de alguns genes dos indivíduos são alterados a fim de proporcionar variedade na população. No contexto de AGs, na etapa **Cruzamento** são aplicados operadores de reprodução, enquanto na etapa **Mutação** são aplicados operadores de mutação.

Esses novos indivíduos resultantes dos operadores de cruzamento e mutação são inseridos em uma nova população a ser avaliada na geração seguinte. Esse processo ocorre na etapa **Atualização**. Em geral, a nova população permanece com o mesmo número de indivíduos da geração anterior, bem como o tamanho dos cromossomos permanecem inalterados.

Por fim, na etapa **Critério de Parada** as condições de parada são verificadas. Caso os critérios sejam satisfeitos, o processo é encerrado. Caso contrário o processo retorna a etapa **Avaliação**. Geralmente os critérios de parada variam bastante com o problema podendo ser desde de o número máximo de gerações até o grau de convergência da função de aptidão em análise.

De acordo com o problema, modela-se uma solução utilizando uma estrutura de dados associada que atua com papel similar ao de um cromossomo (ou indivíduo³) presente no processo de evolução biológica. Essa representação através de indivíduos é característico do algoritmo, uma vez que a região de trabalho é uma versão parametrizada do espaço de busca (CARDOSO, 1998).

² Nomenclatura encontrada na literatura que também se refere a função de aptidão.

³ Cada indivíduo pode ser composto por um ou mais cromossomos. Neste trabalho o indivíduo é composto por um único cromossomo, portanto, neste caso não há distinção quanto a utilização de qualquer um dos dois termos.

Em Algoritmos Gen éticos, inicialmente gera-se de forma aleat ória uma populac ão inicial de indiv íduos candidatos (ou soluç ões) que ser ão evolu ídas em direc ão às melhores soluç ões por meio de um processo de selec ão, mais especificamente, através de uma func ão de aptid ão. Esse processo ocorre por *gerac ões*, ciclos que compreendem a execuç ão das etapas presentes na área hachurada da Figura 7. Nesta mesma populac ão, cada indiv íduo é formado por um conjunto de genes que cont ém o material gen ético e que podem ser alterados via mutac ão ou combinados com algum outro a cada geraç ão por meio de reproduç ão.

O tamanho de uma populac ão influencia diretamente na variedade gen ética da soluç ão. Caso a populac ão seja pequena demais, o algoritmo pode apresentar um baixo desempenho visto que a populac ão abrange um pequeno espac,o de busca do problema. Contudo, um número muito elevado de indiv íduos em uma populac ão, resulta em um processamento muito custoso computacionalmente, tornando o processo em uma busca exaustiva pela melhor soluç ão. Na literatura, pode-se encontrar desde novas abordagens para utilizaç ão de populaç ões de tamanho variável (MINETTI; ALFONSO, 2005; ARABAS; MICHALEWICZ; MULAWKA, 1994; QASIM; ELDOS, 2007) até trabalhos que estudam a influ ência do tamanho da populac ão em relaç ão a velocidade de evoluç ão e qualidade das soluç ões (HU; HARDING; BANZHAF, 2010; LI; KIRLEY, 2002).

A escolha certa sobre a forma de representar um indiv íduo é uma tarefa crucial, uma vez que essa escolha está diretamente relacionada ao tipo de problema (KUMAR, 2013). Quanto à sua representaç ão, a forma mais comum consiste em um indiv íduo constituído de valores binários. Cada gene, nesse caso, tem o valor 0 ou 1. Contudo, outros tipos de codificaç ão também são possíveis, como mostrado nos trabalhos de Kumar (2013) e Aggarwal, Garg e Goswami (2014).

Na Tabela 2 s ão apresentados alguns dos modelos de representaç ão de indiv íduos que podem ser encontrados na literatura.

Tabela 2: Exemplos de representaç ão de indiv íduos.

Tipo de Codificaç ão	Exemplo de indiv íduo codificado	Conteúdo
Binária	0 1 0 0 1 1 0 0	Somente 0s e 1s
Octal	0 3 1 4 5 6 7 0	Representaç ão numérica <i>base-8</i> (0-7)
Hexadecimal	9 7 A E F 9 4 3	Representaç ão numérica <i>base-16</i> (0-9, A-F)
Inteira ³	3 4 1 9 6 7 5 8	Números inteiros
Por Valores	0,12 0,78 0,11 0,73	Números reais, caracteres ou objetos

³ Também conhecida por codificaç ão Por Permutaç ão.

Outra característica que pode ser verificada em AGs é o *elitismo*. Indivíduos da geração corrente que apresentam melhores valores no *rank*, são considerados mais aptos e assim sobrevivem para próxima geração.

Portanto, o algoritmo de funcionamento básico de um Algoritmo Genético pode ser apresentado como:

Algoritmo 1 Algoritmo Genético típico

Entrada: t_{max} : número máximo de gerações, M : tamanho da população

Saída: melhor solução

```

1:  $t \leftarrow 0$ ; // Inicia o contador de gerações
2:  $P(t) \leftarrow \text{iniciaPopulacao}(M)$ ;
3: para  $i \leftarrow 1$  até  $\text{length}(P(t))$  fazer
4:    $P.\text{avaliacao}()$ ; // Avalia a função de aptidão para cada indivíduo de  $P(t)$ 
5: fim para
6: repetir
7:    $P_{new} \leftarrow P.\text{selecao}()$ ;
8:    $P_{new} \leftarrow P.\text{cruzamento}()$ ;
9:    $P_{new} \leftarrow P.\text{mutacao}()$ ;
10:   $t \leftarrow t + 1$ ;
11:   $P_{new}.\text{avaliacao}()$ ;
12:   $P \leftarrow P_{new}$ ;
13: até  $P_{new}.\text{criterioDeParada}(t_{max})$ 
14: return  $P_{new}.\text{melhorSolucao}()$ 

```

Tanto o método utilizado na etapa 7 como os operadores genéticos das etapas 8 e 9 são responsáveis por estabelecer condições que garantem uma evolução eficaz dos indivíduos de uma geração t para a geração seguinte $t + 1$. Os métodos e operadores mais utilizados e que julga-se importante mencionar pela sua utilização neste trabalho, serão descritos nas seções seguintes.

3.2.1 Métodos de Seleção

Este procedimento é responsável pela seleção de indivíduos em uma população para serem pais de novos indivíduos de uma geração seguinte. Em termos gerais, pode-se dizer que tal procedimento opera de forma determinística, uma vez que não ocorre de forma aleatória, e sim através da qualidade da solução que eles refletem.

Dentre as diversas formas de seleção podem-se citar:

- **Seleção por roleta**

Basicamente seu funcionamento resume-se a uma roleta fictícia cujo tamanho de suas

divisões é feita com base na aptidão relativa f_r de cada indivíduo em relação à população, conforme apresentado a seguir

$$f_r(i) = \frac{f_a(i)}{\sum_{j=1}^M f_a(j)}, \quad (3.1)$$

em que $f_a(i)$ é o valor de aptidão absoluta do i -ésimo indivíduo, $\sum_{j=1}^M$ é o somatório das aptidões de todos os indivíduos da população e M é o tamanho da população.

Neste procedimento, decide-se pela utilização ou não do indivíduo no processo de formação da população da geração seguinte de acordo com indivíduos que apresentam altos valores de aptidão (indivíduos que ocupam maiores porções ou fatias da roleta). A roleta é então girada r vezes, dependendo do tamanho da população, e os indivíduos sorteados são selecionados para compor os pais responsáveis por gerar os filhos da geração seguinte.

Conforme exposto em Mitchell (1998), este método pode ser implementado da seguinte maneira:

Algoritmo 2 Seleção por Roleta

Entrada: M : tamanho da população,

Saída: Indivíduos selecionados $\rightarrow Ind_s$

```

1:  $T \leftarrow \sum_{j=1}^M f_a(j)$ ; // Soma dos valores de aptidão relativa
2:  $m \leftarrow M$ 
3: enquanto  $m > 0$  fazer
4:    $r \leftarrow \text{rand}(0, T)$ ; // Gera um número aleatório cujo valor está compreendido entre  $[0, 1]$ 
5:    $S \leftarrow 0$ ;
6:   lac, o 1
7:     para  $i \leftarrow 1$  até  $M$  fazer
8:        $S \leftarrow S + f_a(i)$ ; // Acumula os valores de aptidões relativas de indivíduos percorridos
9:       se  $S \geq r$  então
10:         $Ind_s(i) \leftarrow \text{Seleciona o indivíduo corrente}$ ;
11:        break lac, o 1;
12:       fim se
13:        $i \leftarrow i + 1$ ;
14:     fim para
15:   fim lac, o 1
16:    $M \leftarrow M - 1$ ;
17: fim enquanto

```

Este método apresenta inconvenientes como baixa diversidade da população e convergência prematura devido a alta variância e a forte dependência da escala da função

de aptidão (CASTRO, 2001). Ou ainda, a alocação de indivíduos de baixa aptidão nos casos de populações relativamente pequenas (MITCHELL, 1998).

Diante disso, Baker (1987) propôs um método de amostragem que apresenta uma grande diferença quando comparado ao método da roleta tradicional. Ao invés de girar a roleta r vezes, gira-se apenas uma vez mas com P_r ponteiros igualmente espaçados que são utilizados para selecionar os P_r pais.

• Seleção por torneio

Inicialmente um grupo de indivíduos é selecionado de forma aleatória e com igual probabilidade de escolha independente do grau de adaptação que cada um representa. Em seguida, esses indivíduos passam por um procedimento de torneio no qual os mais aptos são selecionados para serem pais na próxima geração.

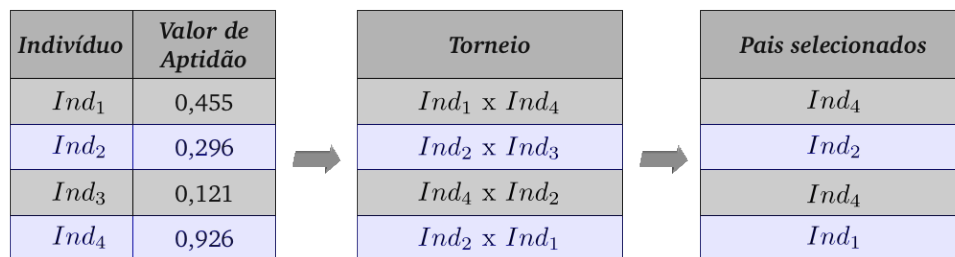


Figura 8: Neste método de seleção, Ind_T indivíduos ($Ind_T \geq 2$) da população são escolhidos aleatoriamente com igual probabilidade. Em seguida, é realizado um torneio no qual o indivíduo mais apto é selecionado.

• Seleção por classificação (ou ranking)

Como já dito anteriormente, utilizar apenas o valor de aptidão relativa para definir o tamanho de divisões de uma roleta, pode resultar em uma alocação insatisfatória, a qual desfavorece ainda indivíduos com baixos valores de aptidão (CASTRO, 2001).

Neste método, o primeiro passo a ser realizado consiste na ordenação dos indivíduos de forma crescente em função do seu valor de aptidão relativa. O passo seguinte é caracterizado pela atribuição de um valor de adequação (ou ordem) para cada indivíduo conforme sua classificação na ordenação. Como exemplo, ao considerar os valores de aptidão relativa dos indivíduos apresentados na Figura 8, após o procedimento de ordenação obtém-se a atribuição do valor de adequação apresentado na Tabela 3.

Como mostrado na Figura 9, após a atribuição dos valores de adequação, o tamanho das divisões da roleta passa a ser melhor distribuída. Neste exemplo, pode-se notar que o indivíduo Ind_2 , passou a ter uma distinção menor em relação ao indivíduo Ind_3 após a atribuição do valor de adequação. Essa falta de vantagem de indivíduos considerados

bons, pode ser facilmente contornada pelo uso da técnica de *elitismo* combinada ao método de seleção.

Tabela 3: Atribuição do valor de adequação para indivíduos no método de Seleção por torneio.

Valor de Aptidão	f_r	Valor de Adequação
0,455	0,253	3
0,296	0,164	2
0,121	0,067	1
0,926	0,515	4

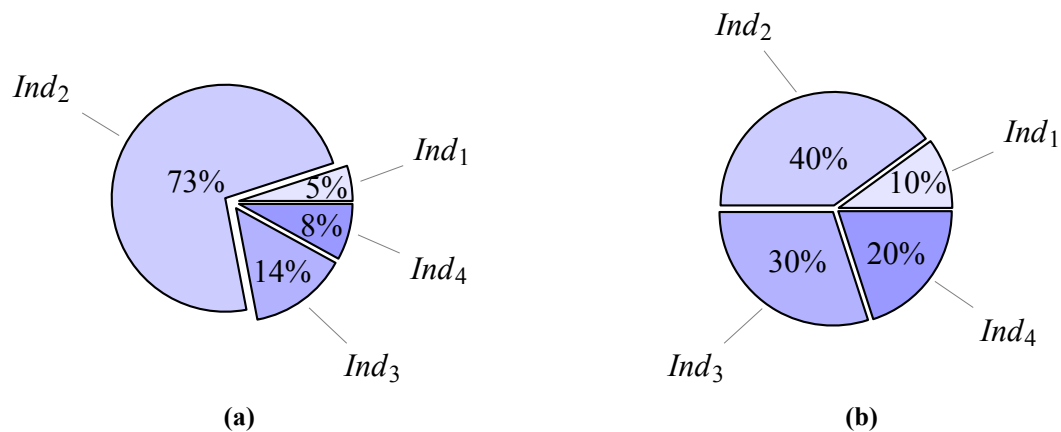


Figura 9: Comparação entre as porções destinadas a indivíduos em uma roleta antes e após a consideração do valor de adequação. (a) distribuição da roleta em função do valor de aptidão; (b) distribuição da roleta após a consideração do valor de adequação.

3.2.2 Operadores de Cruzamento (ou Reprodução)

Este é um poderoso mecanismo de recombinação de soluções que permite a exploração de áreas desconhecidas do espaço de busca, pois sua função consiste na troca de material genético entre indivíduos previamente selecionados por algum método de seleção como os descritos na Seção 3.2.1. O uso destes operadores está diretamente relacionado ao processo de reprodução presente no contexto de Algoritmos Genéticos, de tal maneira que os novos indivíduos gerados herdam as melhores características de determinados indivíduos.

Segundo Malaquias (2006), para a escolha desse operador deve-se levar em consideração um que resulte na solução mais próxima à ótima, considerando ainda uma convergência em um número de iterações razoáveis. Dessa forma espera-se aumentar a qualidade média da população pelo uso desse tipo de operador.

A cada geração, indivíduos mais capazes são originados mantendo características de adaptação das gerações anteriores, uma vez que parte do material genético de um indivíduo é trocado com outro. Por isso, esses novos indivíduos, são potencialmente melhores que os indivíduos utilizados como base nesse tipo de operação.

Este operador faz parte do processo de reprodução verificado em AGs, nos quais novos pares de indivíduos são gerados a partir de pares de pais selecionados. A seguir é apresentada uma lista contendo os operadores de cruzamento mais utilizados nas implementações de Algoritmos Genéticos (GOLDBERG, 1989).

Inicialmente são apresentados os operadores de cruzamento para indivíduos binários, entre eles: cruzamento de único ponto, cruzamento multiponto e cruzamento por ponto uniforme. Em seguida, é apresentado um operador de cruzamento para indivíduos com codificação inteira: *order crossover (OX1)*.

- **Cruzamento de único ponto**

Inicialmente, escolhe-se de forma aleatória um ponto (gene) de separação no indivíduo qualquer p_q dentro do intervalo $[1, < tamanho\ do\ indivíduo - 1 >]$. Em seguida, realiza-se a divisão dos indivíduos envolvidos no processo de reprodução em 2 segmentos de acordo com o ponto p_q escolhido.

Ao se aplicar o operador de cruzamento de único ponto nos indivíduos *Pai 1* e *Pai 2*, tem-se:

- Um novo indivíduo *Filho 1* composto pelo primeiro segmento do cromossomo *Pai 1* concatenado ao segundo segmento do cromossomo *Pai 2*;
- De maneira análoga, um novo indivíduo *Filho 2* é gerado a partir da concatenação do primeiro segmento do cromossomo *Pai 2* concatenado ao segundo segmento do cromossomo *Pai 1*.

Na Figura 10 é ilustrado o processo de geração de filhos usando este operador.

- **Cruzamento multiponto**

Seu funcionamento apresenta-se bastante similar ao *cruzamento de único ponto*. Entretanto, para este operador, k pontos de separação p_q nos indivíduos são escolhidos aleatoriamente, respeitando o tamanho dos indivíduos. Esses k segmentos são trocados de forma alternada entre os indivíduos envolvidos no processo de reprodução. Na Figura 11 tem-se uma ilustração do funcionamento para o operador de cruzamento multiponto.

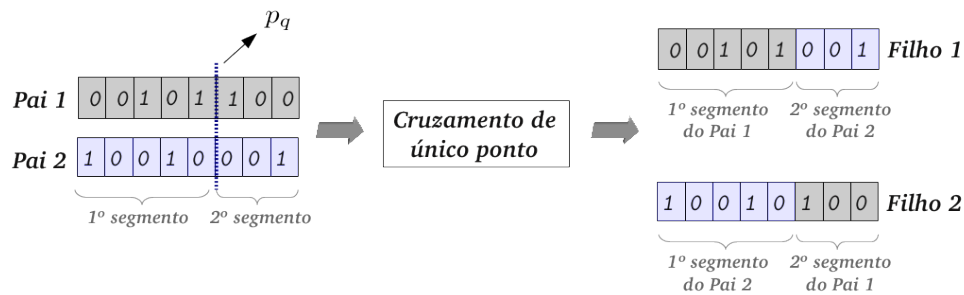


Figura 10: Processo de reprodução através do operador de cruzamento de único ponto.



Figura 11: Processo de reprodução através do operador de cruzamento multiponto.

• Cruzamento por ponto uniforme

Neste tipo de operador, faz-se o uso de uma máscara binária cuja função é definir quais genes serão utilizados na criação dos filhos.

O esquema mostrado na Figura 12 descreve como a máscara binária é utilizada na definição dos genes que serão utilizados na composição de cada filho. Em uma primeira etapa, conforme Figura 12a, para a formação do *Filho 1* cada bit 0 da máscara está associado aos genes do *Pai 1* e consequentemente, cada bit 1 associado aos genes do *Pai 2*, cada um em suas respectivas posições.

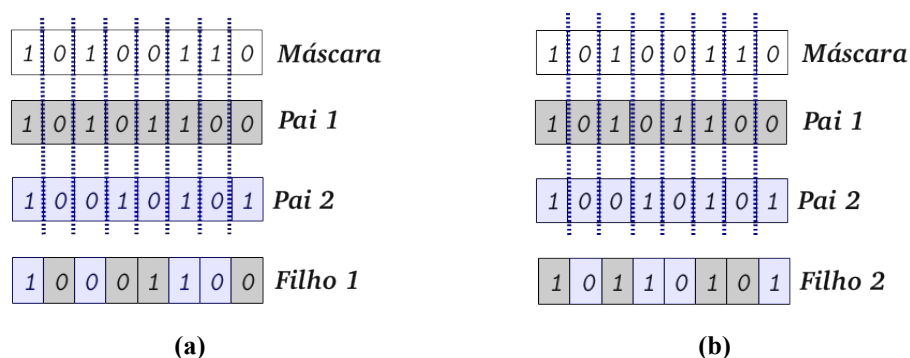


Figura 12: Demonstração do procedimento de geração de filhos por Ponto Uniforme: (a) bits da máscara iguais a 0 referenciam genes do *Pai 1*, caso contrário, o material genético em questão será do *Pai 2*; (b) bits da máscara iguais a 0 referenciam genes do *Pai 2*, caso contrário, o material genético em questão será do *Pai 1*.

Em seguida, inverte-se a definição da máscara (veja Figura 12b), de forma que cada bit 0, passa a referenciar os genes do *Pai 2* e cada bit 1 referenciar os genes do *Pai 1*, resultando

então no *Filho 2*.

- **Order Crossover (OX1)**

Conforme descrito em Goldberg (1989), a partir de dois pais (Figura 13a), escolhe-se aleatoriamente um subconjunto de genes que é mapeado para os respectivos filhos, mantendo suas posições para os novos indivíduos. No exemplo mostrado, o subconjunto escolhido é formado pelos genes localizados nas posições 4, 5 e 6 de cada pai, conforme Figura 13b.

Considerando inicialmente o subconjunto do *Pai 1* como $S_{P_1} = \{3, 7, 2\}$, um cromossomo protótipo é criado contendo como material genético do *Pai 2* que não pertencem ao subconjunto S_{P_1} (ver Figura 13c), mantendo as posições originais do cromossomo de origem.

Em seguida, realiza-se uma rotação de bits a esquerda até que o 2º segmento do cromossomo protótipo seja completamente preenchido, conforme resultado mostrado na Figura 13d. Finalmente, o subconjunto S_{P_1} é copiado para as posições restantes do cromossomo protótipo e um novo filho é criado (Figura 13e).

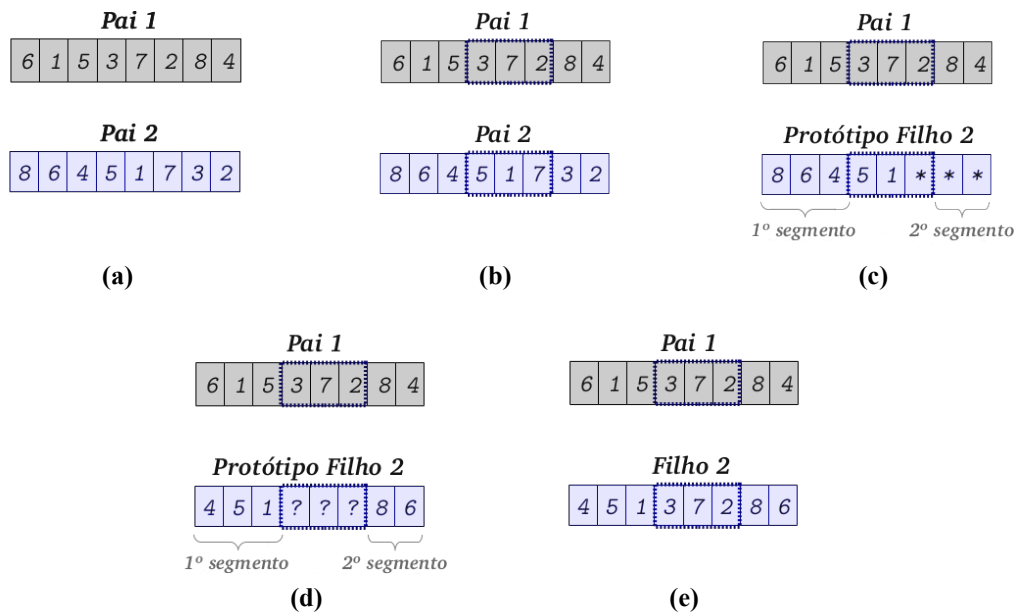


Figura 13: Processo de cruzamento baseado em OX: (a) pais selecionados; (b) escolha aleatória do subconjunto de genes para mapeamento dos filhos; (c) criação do cromossomo protótipo com material genético do *Pai 2* exceto para valores iguais ao subconjunto S_{P_1} ; (d) deslocamento de bits a esquerda para preenchimento dos campos do 2º segmento; (e) cópia do subconjunto S_{P_1} para o protótipo e geração do novo filho.

3.2.3 Operadores de Mutação

Em alguns casos, um Algoritmo Genético pode vir a convergir prematuramente para regiões específicas não desejadas do espaço de busca. Para que essa rápida convergência em regiões de mínimos locais seja contornada, é comum que uma rotina que garanta a biodiversidade seja adicionada a fim de assegurar uma exploração mais ampla sobre o espaço de busca. Com essa finalidade, tem-se os operadores de mutação.

Os operadores de mutação são os responsáveis por alterar parte do material genético de determinados cromossomos sob uma probabilidade de ocorrência previamente definida. Os principais operadores de mutação são descritos a seguir (GOLDBERG, 1989):

- **Flip**

Neste operador, o material genético do gene escolhido para ser mutado é sorteado dentre os valores válidos possíveis de acordo com a codificação utilizada.

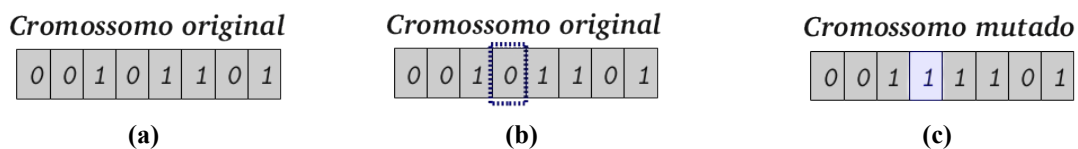


Figura 14: Funcionamento do operador de mutação Flip: (a) cromossomo escolhido para mutação; (b) escolha aleatória do gene a ser mutado em função da probabilidade de mutação; (c) cromossomo mutado.

- **Swap**

No operador Swap pares de genes são sorteados de forma aleatória para realização da troca de material genético entre cada par. Vale ressaltar que a escolha aleatória pode resultar na troca de material genético de valores iguais para o caso de uma codificação binária de cromossomo, tornando-se uma operação irrelevante uma vez que nenhuma alteração no cromossomo é realizada.

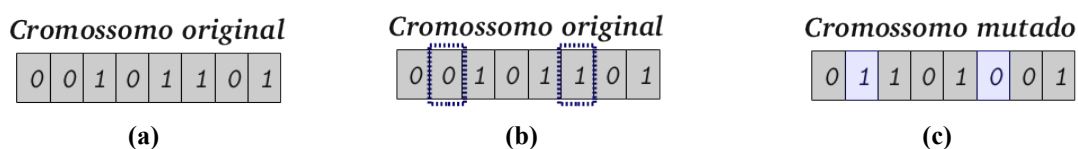


Figura 15: Funcionamento do operador de mutação Swap: (a) cromossomo escolhido para mutação; (b) escolha aleatória do(s) par(es) de troca; (c) cromossomo mutado.

- **Creep**

Neste tipo de operador de mutação, é realizada uma operação de adição ou subtração ao valor do gene escolhido para ser mutado.

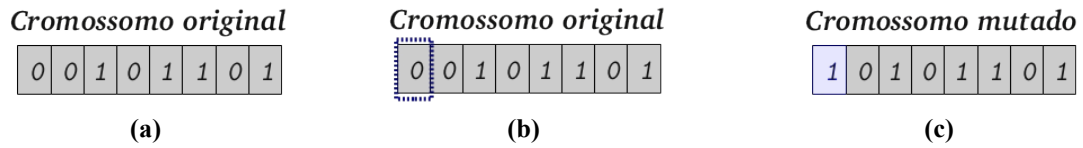


Figura 16: Funcionamento do operador de mutação Creep: (a) cromossomo escolhido para mutação; (b) operação de adição sob o valor do gene escolhido aleatoriamente para ser mutado; (c) cromossomo mutado.

3.3 Otimização Mono e Multiobjetivo com Algoritmos Evolutivos

Otimização pode ser definida como um processo de busca que visa minimizar ou maximizar uma ou mais funções matemáticas (objetivos) podendo ou não estarem sujeitas à algum tipo de restrição que devem ser satisfeitas (BRANKE *et al.*, 2008). Este processo está dividido em duas vertentes a saber: mono-objetivo e multiobjetivo. No caso da otimização mono-objetivo a busca segue apenas uma única função objetivo e geralmente resulta em uma única solução (FREITAS *et al.*, 2010; BRANKE *et al.*, 2008).

Encontrar a solução de problemas reais, muitas vezes consiste na otimização de uma coleção de objetivos, visto que a observação de um único objetivo resultaria em uma abordagem insuficiente para encontrar uma solução que satisfaca o problema em questão. Muitas vezes esses objetivos são conflitantes entre si, de tal forma que não há como melhorar um sem degradar o outro (CASTRO, 2001).

Em virtude da dificuldade na obtenção de soluções ótimas pelos métodos convencionais de otimização, a utilização de algoritmos evolutivos em otimização multiobjetivo consolidou-se de forma eficiente (BARBOSA; RIBEIRO; ARANTES, 2010). Nas últimas décadas, diversas meta-heurísticas foram introduzidas para resolver esse tipo de problema, dentre eles destacam-se:

- *Vector Evaluated Genetic Algorithm - VEGA* (SCHAFER, 1985);
- *Multiple Objective Genetic Algorithm - MOGA* (FONSECA; FLEMING, 1993);
- *Non-Dominated Sorting Genetic Algorithm - NSGA* (SRINIVAS; DEB, 1994);
- *Niched-Pareto Genetic Algorithm - NPGA* (HORN; NAFPLIOTIS; GOLDBERG, 1994);
- *Pareto-Archived Evolutionary Algorithm - PAES* (KNOWLES; CORNE, 1999);
- *Multi-Objective Messy Genetic Algorithm - MOMGA-I* (VELDHUIZEN, 1999);

- *Elitist Non-Dominated Sorting Genetic Algorithm - NSGA-II* (DEB *et al.*, 2000);
- *Rudolph's Elitist Multi-Objective Evolutionary Algorithm - REMOA* (RUDOLPH, 2001);
- *Strength Pareto Evolutionary Algorithm - SPEA2* (ZITZLER; LAUMANN; THIELE, 2001);
- *Multi-Objective Micro-Genetic Algorithm - Micro-GA* (COELLO; GREGORIO, 2001), entre outros

Essas meta-heurísticas geralmente seguem um dos dois principais paradigmas de resolução voltados a esse tipo de problema. O primeiro paradigma utiliza o conceito de Pareto-ótimo, que introduz uma relação de dominância e não-dominância entre os objetivos e assim explora o espaço de busca. Já o segundo paradigma consiste em abordar vários problemas em um único objetivo para encontrar diversas soluções que são não-dominadas pelo problema original (DUBOIS-LACOSTE; LÓPEZ-IBÁÑEZ; STÜTZLE, 2013).

Por sua vez, este trabalho apresenta contribuições que têm por base o segundo paradigma, por isso o detalhamento da outra abordagem apresenta-se fora do escopo desta dissertação. Como será visto na Seção 4.3.2, a função de aptidão do algoritmo proposto baseia-se neste referido paradigma de otimização multiobjetivo.

3.3.1 Hierarquia de Técnicas

Na otimização multiobjetivo tem-se como resultado mais de uma solução que pode ser considerada ótima. Dessa forma se faz necessário o uso de algum procedimento para escolher uma única solução dentre as presentes (BRANKE *et al.*, 2008). Denominado por marcador de decisão (*decision marker, MD*), a partir de determinadas informações de preferências de articulação (*a priori, progressiva, a posteriori*), pode-se escolher implicitamente uma solução aceitável em um problema de otimização multiobjetivo. De acordo com o processo de articulação das preferências do marcador de decisão, os métodos de otimização podem ser classificados conforme divisão mostrada na Figura 17 (VELDHUIZEN; LAMONT, 2000).

Esta hierarquia pode ser mais formalmente definida como:

- **MD com preferência de articulação *a priori*** (decidir $\rightarrow$ pesquisar):

O MD intervém antes do processo de busca da solução, combinando os diferentes objetivos em uma função custo escalar. Essa característica transforma um problema multiobjetivo em um problema de apenas um único objetivo.

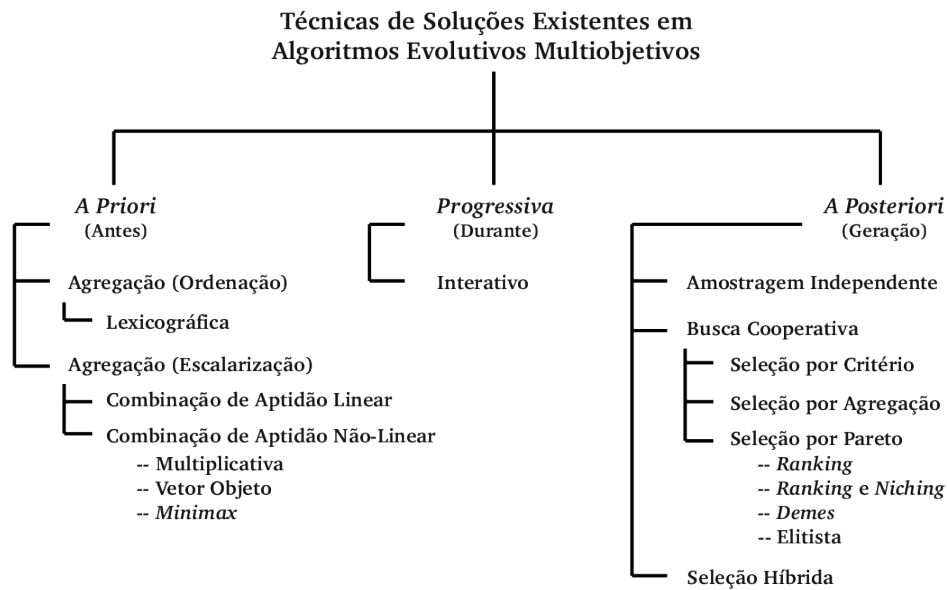


Figura 17: Hierarquia de técnicas existentes em Otimização Multiobjetivo. Traduzido de Veldhuizen e Lamont (2000).

Uma outra forma consiste em definir prioridades entre os objetivos e classificá-los de forma decrescente. Em seguida, resolve-se o primeiro objetivo sem considerar os demais. Dessa mesma forma, o segundo objetivo é otimizado sujeito ao valor ótimo encontrado no primeiro e assim sucessivamente até que todos os objetivos tenham sido resolvidos.

- **MD com preferência de articulação progressiva** (pesquisar $\leftarrow \rightarrow$ decidir):

Geralmente menos estudado que os demais tipos de classificação, este tipo caracteriza-se pela interação entre o processo de busca da solução e a articulação das preferências do MD juntamente com a otimização. Essa interação, atualiza um conjunto de soluções para o MD que direciona a busca para regiões de interesse.

- **MD com preferência de articulação a posteriori** (pesquisar $\rightarrow$ decidir):

A tomada de decisão só acontece após o processo de busca de soluções. Todos os objetivos possuem a mesma importância, logo, com base nas soluções não-dominadas, o MD escolhe a solução que mais se adequa ao problema.

3.3.2 Otimização Multiobjetivo com base em Escalarização

Algoritmos com base em escalarização são utilizados para encontrar diversas soluções não-dominadas para problemas multiobjetivos através da resolução de vários problemas de um único objetivo. Para isso, agregam-se múltiplas funções objetivos em uma única função escalar. Essa agregação transforma o problema multiobjetivo em apenas um, processo esse intitulado

simplesmente por *escalarização* (DUBOIS-LACOSTE; LÓPEZ-IBÁÑEZ; STÜTZLE, 2013).

Existem muitas maneiras de escalarizar problemas multiobjetivos. De acordo com Ehrgott (2005), os métodos de escalarização encontrados na literatura podem ser divididos nas seguintes classes: *Métodos de ponderação*; *Métodos de Restrições*; *Métodos de Pontos de Referência* e *Métodos com base em Direção*.

No entanto, é comum a utilização de poucos métodos padrões em virtude de suas simplificações e propriedades desejadas. Os métodos mais comuns utilizados neste contexto são: *Agregação Linear*; *Agregação Tchebycheff* e *Ordenação Lexicográfica* (DUBOIS-LACOSTE; LÓPEZ-IBÁÑEZ; STÜTZLE, 2013). Outras possibilidades podem ser encontradas ainda nos trabalhos de Ehrgott (2005) e Coello, Lamont e Veldhuizen (2007).

A resolução escalarizada de problemas ocorre geralmente quando o problema multiobjetivo é abordado *a priori*, onde dessa forma, o MD é capaz de definir as características da função escalar auxiliar antes da otimização. Como vantagens atribuídas ao uso de algoritmos escalarizados, pode-se citar o fato da utilização de qualquer algoritmo conhecido para resolver problemas de escalarização eficientemente. Contudo, as soluções são encontradas uma por vez e o número total de soluções não-dominadas retornadas é no máximo o número de escalarizações consideradas (DUBOIS-LACOSTE; LÓPEZ-IBÁÑEZ; STÜTZLE, 2013).

3.4 Conclusão

Algoritmos Genéticos são caracterizados pela capacidade de evolução proporcionada pelo uso de operadores inspirados na genética e na seleção. O fato de possuírem uma população composta por M indivíduos, significa que realizam uma procura simultânea em vários espaços de busca durante o processo de evolução. Sendo assim, a diversidade das soluções encontradas evita que o algoritmo possa vir a convergir a mínimos locais.

O tipo de codificação a ser utilizada, assim como o tamanho da população, estão diretamente interligados às especificações do problema de otimização a ser resolvido. Logo, a escolha incorreta dessas representações pode ocasionar em resultados insatisfatórios ou muito custosos computacionalmente.

Problemas reais podem se caracterizar pela solução de uma única função objetivo (otimização mono-objetivo) ou várias funções objetivos (otimização multiobjetivo), conflitantes entre si ou não. Conforme mencionado anteriormente, técnicas de otimização multiobjetivo resultam em diversas soluções viáveis, fazendo-se necessário o uso de um marcador de decisão

para escolher a melhor solução. Este por sua vez pode intervir antes ou depois do processo de busca da solução ou ainda, agir em conjunto com a função de decisão direcionando a busca para regiões específicas de interesse.

MÉTODOS PARA REDUÇÃO DE VETORES -S UPORTE

A falta de esparsidade é um inconveniente não verificado nos classificadores SVM que acomete os modelos LSSVM. Falar sobre imposição de esparsidade em LSSVM consiste na utilização de algoritmos de poda, que descartam vetores-suporte que apresentam pouca relevância para a solução do sistema. Neste capítulo é apresentada uma breve descrição da evolução dessas soluções. Apresenta-se também, três técnicas encontradas na literatura que são utilizadas para fins de comparação com os métodos propostos nesta dissertação.

Em seguida, propõe-se dois novos métodos para obtenção de um conjunto reduzido de vetores-suporte por meio de poda daqueles vetores menos significativos, ambas as soluções propostas têm por base o uso de Algoritmos Genéticos. Ressalta-se que o uso de AGs neste trabalho tem propósito diferente ao normalmente encontrado na literatura, em que seu uso restringe-se a encontrar os valores ótimos dos hiperparâmetros do classificador. Nos métodos propostos, AGs são usados para imposição de esparsidade aos multiplicadores de Lagrange.

4.1 Introdução

O classificador LSSVM é uma alternativa à formulação da Máquina de Vetores-Suporte padrão (SUYKENS; LUKAS; VANDEWALLE, 2000b). Nessa formulação, há duas importantes modificações quando comparada à formulação do classificador SVM. Como consequência, tem-se que o treinamento da Máquina de Vetores-Suporte por Mínimos Quadrados se dá pela resolução de um sistema de equações lineares ao invés de um problema de otimização quadrática.

Uma das principais características dos classificadores SVM, está na detecção automática de vetores-suporte que auxiliam na construção do hiperplano de separação. Esses vetores localizam-se sobre a fronteira de decisão entre as classes e são obtidos após o processo

de treinamento do classificador é caracterizado pela esparsidade do vetor de multiplicadores de Lagrange. Como muitos deles são nulos, apenas uma pequena porção do conjunto de treinamento é utilizada como vetores-suporte, consequentemente, apenas essa pequena porção de elementos deve ser armazenada para avaliação de padrões não vistos na etapa de teste do classificador.

Diferentemente do verificado nos classificadores SVM, a incapacidade de detecção automática de vetores-suporte que acomete os modelos LSSVM se dá pela perda de esparsidade da solução obtida. Conforme discussão apresentada na Seção 2.4.2, raramente existem α_i nulos e consequentemente, todos os elementos do conjunto de treinamento contribuem para compor a superfície de decisão do classificador.

A detecção automática de vetores-suporte é uma característica desejada devido sua contribuição em diversos aspectos em um classificador. Lee e Mangasarian (2001), elencam como motivação para redução, dificuldades na utilização de kernels não lineares em grandes conjuntos de dados, o que resulta em um alto consumo de memória. Além disso, é exigido um grande custo computacional para o processamento matemático na obtenção da superfície de separação. O algoritmo proposto por Lee e Mangasarian (2001), mostra que é possível representar uma base de dados inicialmente grande, por meio de poucos vetores que garantam tal representatividade.

Hady *et al.* (2011) citam que um grande número de elementos sendo vetores-suporte, pode resultar em um super ajuste e consequentemente o classificador pode se adaptar ao ruído existente no conjunto de treinamento e não a verdadeira distribuição dos dados. Isso faz com que ocorram falhas no momento em que o classificador é utilizado para categorizar padrões não vistos no treinamento.

Além disso, a constante evolução de tecnologias de coleta de dados, têm contribuído para o crescimento exponencial na quantidade de dados armazenados (CAMARGO, 2010). Ainda neste raciocínio, verifica-se atualmente um grande desafio computacional na análise eficiente de grandes volumes de dados devido a alta dimensionalidade e esparsidade verificadas. Para Rodrigues (2011), características como as citadas anteriormente são responsáveis por motivar o desenvolvimento de algoritmos capazes de manter um bom desempenho na análise desse tipo de dados.

O desenvolvimento de algoritmos que impõem esparsidade têm por base critérios que determinam quais vetores do conjunto de treinamento podem ser considerados vetores-suporte. Através desses critérios, vetores que representam pouca significância para a construção do hiperplano de separação, são eliminados do conjunto de treinamento e não necessitam ser

armazenados para a etapa de teste. Esses vetores podem ser escolhidos tanto durante o procedimento de treinamento do classificador como em etapas posteriores (CARVALHO, 2005).

Esses métodos têm particular importância para lidar com grandes conjuntos de dados quando um grande número de padrões estão propícios a serem selecionadas como vetores-suporte. Isso pode ocorrer por perda de vetores-suporte menos relevantes (GEEBELEN; SUYKENS; VANDEWALLE, 2012; LI; LIN; ZHANG, 2006) ou através de um conjunto reduzido de exemplos de treinamento (PERES; PEDREIRA, 2009; CARVALHO; BRAGA, 2009), em muitos casos apresentando um impacto mínimo do desempenho.

4.2 Classificadores Esparsos

Muitos trabalhos foram desenvolvidos com o intuito de impor esparsidade em Máquinas de Vetores-Suporte por Mínimos Quadrados, dentre eles, segundo Carvalho e Braga (2009), destacam-se métodos como Pruning LSSVM e LS^2 - SVM. A partir desses métodos, vários outros foram propostos tomando-os como base de inspiração.

O método Pruning LSSVM (ou P-LSSVM) proposto por Suykens, Lukas e Vandewalle (2000b) é um processo iterativo que requer a solução de um conjunto de equações¹ a cada etapa, o que multiplica a complexidade (VALYON; HORVATH, 2004). Uma solução considerada ótima, deve combinar 3 características desejáveis: deve ser rápida, deve conduzir a uma solução esparsa e deve ser capaz de produzir resultados equiparáveis aos obtidos em modelos clássicos.

Proposto em função das considerações expostas anteriormente, o método LS^2 - SVM (VALYON; HORVATH, 2004), possui duas etapas definidas no processo de remoção automática de vetores de treinamento. A primeira etapa é responsável pela redução da matriz $\mathbf{D}$ através de operações matemáticas que a transformam em sua forma escalonada reduzida. Em seguida, é realizado um *lac*o sobre a matriz (índice linha i , índice coluna j), a fim de determinar o maior elemento p da coluna j . Se $p \leq \varepsilon$ (onde ε é um limiar previamente definido) então essa parte da matriz recebe valores nulos. Este método retorna uma lista de vetores coluna que são linearmente independentes dos outros, considerando o limiar ε . A escolha de ε assemelha-se a seleção dos hiperparâmetros da SVM.

Vale ressaltar que somente as colunas com valores iguais a zero são removidas, o que resulta em uma matriz não quadrada. Logo, a segunda etapa consiste na resolução do novo sistema com base no método da função Pseudo-Inversa, já descrito na Seção 2.5.2.

¹ Cujos tamanho é reduzido de forma gradual.

Outro método cuja estratégia de imposição de esparsidade também é dividida em duas etapas definidas, chama-se *Ada-Pinv* (CARVALHO, 2005). Neste método a resolução do sistema linear apresentado na Eq. (2.36) é resolvido mediante a utilização de uma rede neural *Adaline*. Seu uso se faz necessário uma vez que métodos iterativos clássicos não são capazes de resolver esse tipo de sistema por não serem definidos positivo.

Na segunda etapa, vetores que possuem os menores valores de erro de treinamento são eliminados. A porção de vetores-suporte que se deseja obter é representada pelo termo τ .

Um outro método denominado *RRS-LSSVM* (CARVALHO; WILIAN; ANTONIO, 2007) baseia-se em uma modificação da regra do vizinho mais próximo, que leva em consideração a distância entre elementos de classes opostas, para determinação de um subconjunto de padrões de treinamento a serem considerados vetores-suporte. Uma vez identificados, esses padrões são utilizados pelo classificador LSSVM para definir a superfície de decisão entre as classes.

Já o método *Opposite Maps* (ROCHA NETO; BARRETO, 2013) inicialmente realiza a divisão do conjunto de treinamento em dois subconjuntos, um para cada classe. O próximo passo consiste em treinar uma rede SOM (*Self-Organizing Map*) para cada subconjunto definido. Após definir os neurônios vencedores (*Best Matching Unit, BMU*) e podar os neurônios restantes (ou neurônios mortos), o conjunto de dados de uma referida classe é apresentada à rede SOM da classe oposta e novamente os neurônios vencedores correspondentes são definidos. Para cada neurônio ativado anteriormente, encontram-se K pontos mais próximos. Neste momento, o conjunto reduzido de vetores candidatos a pertencerem ao conjunto de vetores-suporte é encontrado.

A seguir, serão apresentados em detalhes dois métodos utilizados para imposição de esparsidade em Máquinas de Vetores-Suporte por Mínimos Quadrados. O primeiro, *P-LSSVM*, trata-se do método base de inspiração para muitos dos algoritmos propostos com tal finalidade. O segundo, *IP-LSSVM*, mostrou desempenho equivalente ou superior quando comparado a métodos como *LS² - SVM*, *Pruning*, *Ada - Pinv* e *RRS + LSSVM*, conforme pode ser visto em (CARVALHO; BRAGA, 2009).

Ambos reduzem o número de vetores-suporte com base no valor dos multiplicadores de Lagrange. Esses métodos são utilizados neste trabalho para comparação com as técnicas propostas nas Seções 4.3.2 e 4.3.3 que usam Algoritmos Genéticos para alcançar esparsidade.

4.2.1 Pruning LSSVM

A partir da escolha da norma-2 e também pelo fato de os valores de suporte serem proporcionais aos erros nos pontos de dados, conforme verificado na igualdade $\alpha_i = \gamma \xi_i$ da Eq. (2.35), o classificador LSSVM apresenta uma desvantagem em termos de esparsidade, em comparação com a formulação original do classificador SVM (SUYKENS; LUKAS; VANDEWALLE, 2000b).

O método de poda P-LSSVM proposto por Suykens, Lukas e Vandewalle (2000b) elimina vetores-suporte e seus respectivos vetores de entrada com base em uma ordenação prévia dos vetores-suporte a partir dos valores absolutos dos multiplicadores de Lagrange $|\alpha_i|$ realizado de forma iterativa.

Suykens, Lukas e Vandewalle (2000b) propuseram uma remoção gradual em torno de 5% do tamanho do conjunto de vetores-suporte. O critério de parada nesta proposta é associado a diminuição do desempenho do conjunto de validação. Vetores são eliminados através da definição dos multiplicadores de Lagrange próximos a zero, sem alterações nas dimensões da matriz.

Portanto, os seguintes passos são realizados para a devida remoção de padrões menos significantes do conjunto de treinamento:

1. Treinar um classificador LSSVM utilizando todos os padrões do conjunto de treinamento.
2. Remover uma pequena quantidade de padrões (algo em torno de 5%) com os menores valores obtidos na ordenação do espectro de $|\alpha_i|$.
3. Retreinar o classificador LSSVM com base no conjunto de treinamento reduzido.
4. Retornar ao passo 2 até que o desempenho do conjunto de validação diminua.

Realiza-se a cada iteração a resolução de um novo sistema linear com base no conjunto reduzido obtido na iteração anterior. Sistemas desse tipo são ditos multipasso, uma vez que o sistema linear necessita ser resolvido diversas vezes até que o critério de parada seja atendido (CARVALHO; BRAGA, 2009).

Com o intuito de realizar uma comparação justa com o classificador proposto FSGAS-LSSVM que usa AGs e visa obter um conjunto de vetores-suporte de tamanho fixo proposto neste trabalho, o classificador P-LSSVM sofre uma pequena modificação quando comparado com sua proposição original. O critério de parada passa a ser apenas a redução desejada e não mais a diminuição do desempenho do conjunto de validação.

Isso se faz necessário para que ao se atingir o percentual de redução proporcional ao valor em questão, mesmo sem qualquer alteração no desempenho do conjunto de validação, o procedimento de remoção de vetores seja encerrado e neste momento se tenha os multiplicadores de Lagrange relacionados aos vetores-suporte que representam todo o conjunto de treinamento.

4.2.2 IP-LSSVM

O método de poda IP-LSSVM proposto por Carvalho e Braga (2009) leva em consideração como critério de detecção de vetores-suporte, padrões que se encontram próximos a superfície de separação e padrões muito distantes dos hiperplanos de apoio de cada classe (H_{-1} e H_{+1}).

Este método lança mão da ideia de que padrões com $\alpha_i = 0$ tendem a se tornar vetores-suporte, pois os padrões estão localizados mais próximos da superfície de decisão do que dos hiperplanos de apoio (H_{-1} e H_{+1}). Portanto, essa ideia aplicada no treinamento do método IP-LSSVM tem por base o modo de operação dos classificadores SVM. Os novos critérios de relevância propostos no trabalho de Carvalho e Braga (2009), para se considerar padrões como vetores-suporte, podem ser descritos como:

- $\mathbf{x}_i$ com $\alpha_i = 0$ é considerado vetor-suporte. Neste caso, $\mathbf{x}_i$ é colocado na fronteira entre as duas classes ou na região da classe oposta. Corresponde a vetores associados não nulos com $\alpha_i = 0$ quando comparados aos classificadores SVM solucionados por problemas de programação quadrática.
- $\mathbf{x}_i$ com $\alpha_i \geq 0$ é removido. $\mathbf{x}_i$ é corretamente classificado, perto do hiperplano de apoio, o que corresponde ao vetor associado ao multiplicador de Lagrange com $\alpha_i = 0$ nos classificadores SVM.
- $\mathbf{x}_i$ com $\alpha_i < 0$ é removido. $\mathbf{x}_i$ é corretamente classificado, longe do hiperplano de decisão, o que corresponde ao vetor associado ao multiplicador de Lagrange com $\alpha_i = 0$ nos classificadores SVM.
- $\mathbf{x}_i$ com $\alpha_i = 0$ é removido. Novamente, $\mathbf{x}_i$ é corretamente classificado, longe do hiperplano de decisão, o que corresponde ao vetor associado ao multiplicador de Lagrange com $\alpha_i = 0$ nos classificadores SVM.

O critério proposto é então aplicado ao método IP-LSSVM em duas etapas definidas. O objetivo é eliminar colunas não relevantes da matriz $\mathbf{D}$ e construir uma matriz reduzida não

quadrada (denominada neste trabalho por $\mathbf{D}_2$) a ser utilizada posteriormente. A eliminação de colunas representa na realidade a remoção de vetores menos relevantes para o problema de classificação em questão, selecionados de acordo com seus respectivos valores dos multiplicadores de Lagrange.

A primeira etapa consiste na utilização da inversa para a resolução do sistema de equações lineares representados pela Eq. (2.36). A solução desse sistema é apresentado pela Eq. (2.42). Ressalta-se novamente que onde o primeiro elemento do vetor é o viés b e o mesmo não é considerado, somente os valores de α são de interesse nesta fase de eliminação.

Já a segunda etapa é caracterizada pela resolução de um sistema $\mathbf{D}_2 \mathbf{z}_2 = \mathbf{l}_2$ através do método da Pseudo-Inversa. O processo de treinamento do classificador esparso proposto por Carvalho e Braga (2009) é apresentado a seguir.

1. O sistema de equações lineares representado na Eq. (2.36) deve ser resolvido utilizando todos os vetores de treinamento através de $\mathbf{z} = \mathbf{D}^{-1} \mathbf{l}$, uma vez que $\mathbf{D}$ é uma matriz quadrada.
2. O parâmetro $\tau \in [0, 1]$ é definido. Este parâmetro representa a porcentagem de vetores de treinamento que serão considerados vetores-suporte.
3. Os vetores de treinamento são ordenados de acordo com seus respectivos valores de α .
4. A porcentagem $1 - \tau$ dos dados de treinamento que correspondem aos menores valores são selecionados.
5. A matriz não quadrada $\mathbf{D}_2$ é gerada pela remoção das colunas da matriz $\mathbf{D}$ associadas aos elementos selecionados em α .
6. O novo sistema de equações lineares, representado por $\mathbf{D}_2 \mathbf{z}_2 = \mathbf{l}_2$, é resolvido como $\mathbf{z}_2 = \mathbf{D}_2^\dagger \mathbf{l}_2$, conforme demonstrado na Eq. (2.44).
7. Os pontos de treinamento de $\mathbf{D}_2$ são definidos como vetores-suporte.
8. Por fim, os valores de α 's e b são obtidos de $\mathbf{z}_2$.

O classificador esparso proposto, não possui modificações quanto aos parâmetros de formulação do classificador original LSSVM. Tanto o parâmetro de regularização γ como os parâmetros da função kernel a ser utilizada, devem ser estipulados *a priori*. Contudo, um novo parâmetro τ , que relaciona a porcentagem de vetores que serão considerados vetores-suporte, deve ser informado.

4.3 Classificadores Esparsos com base em AGs

Algoritmos Evolutivos, mais especificamente Algoritmos Genéticos, têm sido amplamente utilizados para resolver problemas de otimização do mundo real nas mais diversas áreas, tais como bioinformática, medicina, economia, química (JAUHARI; RIZVI, 2014; FARAGÓ *et al.*, 2014; ALVAREZ *et al.*, 2012; PALENCIA; DELGADILLO, 2012; BRAIN; ADDICOAT, 2010). Algoritmos Genéticos, são métodos de otimização que podem ser utilizados para gerar soluções adequadas para busca.

De uma forma geral, Algoritmos Genéticos são geralmente aplicados com o intuito de obter os parâmetros ótimos da função de kernel desejada, como também de ajustar os parâmetros específicos do classificador (MUSTAFA *et al.*, 2012; SAMADZADEGAN; SOLEYMANI; ABBASPOUR, 2010; YU *et al.*, 2009). Em função das características subjacentes aos AGs, alguns problemas de otimização podem ser resolvidos sem a suposição de linearidade, diferenciabilidade, continuidade e convexidade da função objetivo, o que difere das características presentes em vários outros modelos matemáticos clássicos quando submetidos ao mesmo tipo de problema.

Diferentemente de trabalhos anteriores, a utilização de AGs neste trabalho visa alcançar soluções esparsas e não descobrir os valores ótimos dos parâmetros de kernel e do classificador. Pesquisas realizadas durante o desenvolvimento deste trabalho, resultaram em apenas um método que impõe esparsidade a classificadores LSSVM através da utilização de Algoritmos Genéticos, denominado de *GAS-LSSVM* (SILVA; ROCHA NETO, 2013).

O método de Silva e Rocha Neto busca encontrar um indivíduo que melhor represente a solução para o sistema linear do classificador LSSVM através do uso de Algoritmos Genéticos Mono-Objetivo em função da acurácia sobre o conjunto de treinamento (ou validação) e ao mesmo tempo impõe esparsidade aos multiplicadores de Lagrange. Tal método será melhor detalhado a seguir.

4.3.1 GAS-LSSVM

Em sua proposta, o classificador *Genetic Algorithms for Sparse LSSVM*, *GAS-LSSVM*, possui indivíduos (ou cromossomos) representados por vetores de genes binários, de tal maneira que cada gene está relacionado a um padrão. Cada gene pode ser definido como “zero”, o que indica que o padrão correspondente é utilizado como vetor-suporte no processo de treinamento, ou “um” caso contrário.

Vale ressaltar que a partir deste momento, neste trabalho, diferentemente da codificação utilizada para o classificador GAS-LSSVM convencionada-se que genes definidos para “um” indicam que o padrão associado é utilizado como vetor-suporte ou “zero”, caso contrário.

A função de aptidão é definida como sendo a taxa de acerto do classificador resultante levando em consideração os genes definidos como “um” (genes que representam os padrões escolhidos para serem vetores-suporte). Assim, o algoritmo é então guiado em função da taxa de acerto encontrada a cada iteração, no entanto, paralelamente, essa abordagem alcança a esparsidade devido a possibilidade de ter ou não um determinado padrão como vetor-suporte.

Similarmente ao que ocorre com o classificador IP-LSSVM, no método proposto por Silva e Rocha Neto (2013), apenas as colunas de interesse são removidas, que neste caso são os multiplicadores de Lagrange definidos como “zero” pela modelagem do AG. Se a *késima* linha é removida, então o padrão $(\mathbf{x}_k, y_k)$ de treinamento é perdido. Isto significa que uma restrição é retirada do problema, o que conduz a uma perda de informação e desempenho na solução obtida (VALYON; HORVATH, 2004).

O algoritmo para o classificador GAS-LSSVM é descrito a seguir.

Algoritmo 3 Algoritmo do Classificador GAS-LSSVM

Entrada: t_{max} : número máximo de gerações, M : tamanho da população

Saída: melhor solução

```

1:  $t \leftarrow 0$ ; // Inicia o contador de gerações
2:  $P(t) \leftarrow \text{iniciaPopulacao}(M)$ ;
3: Construir a matriz  $\{\mathbf{D}_2^i\}_{i=1}^s$ ; //  $s$  : número de indivíduos da geração  $t$ 
4: para  $i \leftarrow 1$  até  $\text{length}(P(t))$  fazer
5:   Remover as colunas da matriz  $\mathbf{D}_2^i$  de acordo com os genes definidos para “zero”;
   // Poda de vetores-suporte
6:    $\mathbf{z}_2^i \leftarrow \mathbf{D}_2^i \mathbf{l}_2^i$ ;
7:   Avaliar a taxa de acerto do conjunto de treinamento; // Função de aptidão
8: fim para
9: repetir
10:   $P_{new} \leftarrow P.\text{selecao}()$ ; // Seleciona as matrizes  $\mathbf{D}_2^i$ 
11:   $P_{new} \leftarrow P.\text{cruzamento}()$ ;
12:   $P_{new} \leftarrow P.\text{mutacao}()$ ;
13:   $t \leftarrow t + 1$ ;
14:  Criar uma matriz  $\mathbf{D}_2^i$  para cada indivíduo em  $P(t + 1)$ ;
15:  Remover as colunas da matriz  $\mathbf{D}_2^i$  de acordo com os genes definidos para “zero”;
   // Poda de vetores-suporte
16:   $P_{new}.\text{avaliacao}()$ ; // Avalia a taxa de acerto para cada  $\mathbf{D}_2^i$ . Função de aptidão
17:   $P \leftarrow P_{new}$ ;
18: até  $P_{new}.\text{critérioDeParada}(t_{max})$ 
19: return  $P_{new}.\text{melhorSolucao}()$ ;

```

4.3.2 MOGAS-LSSVM (Proposta 1)

Como dito anteriormente, neste trabalho busca-se combinar as vantagens de classificadores LSSVM e Algoritmos Genéticos colocando-os para trabalhar em conjunto em novas propostas para redução do número de vetores suporte e também alcançar equivalentes desempenhos (e em alguns casos superiores) quando comparadas ao classificador LSSVM padrão.

O método *Multi-Objective Genetic Algorithm for Sparse LSSVM*, *MOGAS-LSSVM* proposto neste trabalho consiste em uma nova abordagem para a construção de classificadores LSSVM com base em Algoritmos Genéticos Multiobjetivos (*Multi-Objective Genetic Algorithms*, *MOGA*) com escalarização para problemas de classificação de padrões (SILVA; ROCHA NETO, 2014).

Nesta abordagem, o método é capaz de descobrir esparsidade através de um MOGA que leva em consideração tanto o desempenho de classificação (acurácia) quanto a taxa de poda de vetores-suporte. A ideia principal é deixar de fora padrões discrepantes (*outliers*), padrões irrelevantes ou aqueles que podem estar corrompidos com ruído e que assim, impedem o classificador de obter acurácia mais elevada.

O MOGAS-LSSVM tem por objetivo maximizar o desempenho do classificador sob o conjunto de treinamento (ou validação) e minimizar o número de vetores-suporte. Essa ponderação é alcançada através de uma função de aptidão combinada por escalarização (DUBOIS-LACOSTE; LÓPEZ-IBÁÑEZ; STÜTZLE, 2013), de tal maneira que um problema multiobjetivo pode ser transformado em um problema com apenas um objetivo.

A função de aptidão proposta neste trabalho com base em escalarização apresenta um balanço entre o valor resultante do desempenho do classificador e a proporção de vetores-suporte podados.

O classificador MOGAS-LSSVM utiliza os seguintes operadores genéticos, a saber: os indivíduos escolhidos para as gerações seguintes são selecionados através do método de *seleção por roleta*. Na reprodução, o operador utilizado para cruzamento dos pais selecionados foi o *cruzamento multiponto*. Por fim, na mutação, 10% dos genes (de acordo com o tamanho do indivíduo) são mutados de forma aleatória.

4.3.2.1 Indivíduos (ou cromossomos)

Cada indivíduo ou cromossomo é representado por um vetor de genes binários iniciados de forma aleatória. Um padrão é considerado um vetor-suporte quando o gene associado à ele é

igual a “um”. Logo, para um valor igual a “zero” para um gene, o padrão não é vetor-suporte².

Considere uma matriz **D** de um sistema linear LSSVM composta por um pequeno conjunto de treinamento com quatro padrões conforme a seguinte equação:

$$\mathbf{D} = \begin{bmatrix} 0,00 & 1,00 & -1,00 & -1,00 & 1,00 \\ 1,00 & 27,81 & 5,03 & -0,12 & 1,25 \\ -1,00 & 5,03 & 37,36 & -2,29 & 4,12 \\ -1,00 & -0,12 & -2,29 & 28,79 & 0,87 \\ 1,00 & 1,25 & 4,12 & 0,87 & 28,99 \end{bmatrix}. \quad (4.1)$$

Em MOGAS-LSSVM, para todos os padrões definidos vetores-suporte, tem-se um vetor de genes no formato $[1 \ 1 \ 1 \ 1]$. Para um indivíduo sem o quarto padrão pertencente ao conjunto de vetores-suporte, cria-se um vetor de genes no formato $[1 \ 1 \ 1 \ 0]$. Assim, a matriz **D**₂ obtida é representada por:

$$\mathbf{D}_2 = \begin{bmatrix} 0,00 & 1,00 & -1,00 & -1,00 \\ 1,00 & 27,81 & 5,03 & -0,12 \\ -1,00 & 5,03 & 37,36 & -2,29 \\ -1,00 & -0,12 & -2,29 & 28,79 \\ 1,00 & 1,25 & 4,12 & 0,87 \end{bmatrix}. \quad (4.2)$$

Isso significa que a quinta coluna é eliminada, conforme a Eq. (4.2). De um modo geral, a fim de eliminar um certo padrão do conjunto de vetores-suporte que é descrito pelo n -ésimo gene g_n no vetor $\mathbf{g} = [g_1 \ g_2 \ g_3 \ g_4]$, faz-se necessário a remoção do gene $n + 1$.

O processo de evolução dos indivíduos ao longo das gerações ocorre da seguinte maneira: 10% dos indivíduos que apresentam o maior valor de aptidão na geração t , são mantidos por *elitismo* para a geração $t + 1$, 80% dos indivíduos são gerados pelo operador de cruzamento e os indivíduos restantes são obtidos pelo operador de mutação.

Por exemplo, considere uma população composta por 120 indivíduos, para uma nova população, 12 deles são escolhidos por *elitismo*. Dos 108 indivíduos restantes, 86 são gerados através do operador de reprodução, enquanto, 22 novos filhos são obtidos usando o operador de mutação.

As considerações aqui expostas referentes à formulação dos indivíduos no que diz respeito

² Diferentemente do que é visto no trabalho de Silva e Rocha Neto (2014), onde o gene que será utilizado como vetor-suporte no processo de formação recebe o valor “zero” (falso), caso contrário, o gene receberá o valor “um” (verdadeiro).

à representação, convenção de valores utilizados para representar a presença (1) ou ausência (0) de um padrão no conjunto de vetores-suporte, esquema de remoção de colunas, bem como a forma de evolução dos indivíduos ao longo das gerações, também se estendem aos indivíduos do classificador FSGAS-LSSVM que é abordado na Seção 4.3.3.

4.3.2.2 Função de Aptidão

No que diz respeito a modelagem do indivíduo, o classificador MOGAS-LSSVM busca tanto maximizar a acurácia do classificador quanto minimizar o número de vetores-suporte. Contudo, este processo de otimização leva em conta um certo custo de poda de padrões θ que pertencem ao conjunto de vetores-suporte.

Tais características são obtidas através da seguinte função a saber:

$$aptidão = 1 - \frac{(1 - acurácia) + ((1 - \theta) * redução)}{2 - \theta}, \quad (4.3)$$

em que $\theta \in [0, 1]$ é um parâmetro específico do classificador, *acurácia* é a taxa de acerto sobre o conjunto de treinamento (ou validação) e

$$redução = \frac{\#padrões \text{ de treinamento podados}}{\#padrões \text{ de treinamento}}. \quad (4.4)$$

Ao se avaliar a Eq. (4.3) pode-se notar que o valor de aptidão está compreendido entre o intervalo $[0, 1]$, uma vez que se o custo θ é elevado (um), então os valores da função de aptidão somente dependem da *acurácia*. Por outro lado, se o custo θ é baixo (zero), então os valores da função de aptidão além de dependerem da *acurácia*, passam a ter uma proporção direta com a *redução*.

Diante do exposto, sempre que $\theta = 1$, o classificador MOGAS-LSSVM é guiado apenas em função do valor da acurácia, o que resulta em um único objetivo, sendo desta forma, equivalente ao classificador GAS-LSSVM (SILVA; ROCHA NETO, 2013). Logo, o classificador MOGAS-LSSVM proposto neste trabalho apresenta-se mais geral que o classificador GAS-LSSVM.

4.3.2.3 Parâmetros do Classificador MOGAS-LSSVM

No método proposto são utilizados os mesmos parâmetros do classificador LSSVM padrão, que são: μ e os parâmetros da função de kernel (como utiliza-se uma função kernel linear, tal parâmetro não é necessário). No entanto, outros parâmetros particulares ao processo de

otimização do Algoritmo Genético são necessários. Adicionalmente, o custo de poda θ também deve ser definido.

4.3.2.4 Algoritmo MOGAS-LSSVM

O algoritmo para treinar um classificador MOGAS-LSSVM é descrito a seguir.

Algoritmo 4 Algoritmo do Classificador MOGAS-LSSVM

Entrada: t_{max} : número máximo de gerações, M : tamanho da população, θ : custo de poda de padrões

Saída: melhor solução $\rightarrow \mathbf{D}^o$

```

1:  $t \leftarrow 0$ ; // Inicia o contador de gerações;
2:  $P(t) \leftarrow \text{iniciaPopulacao}(M)$ ;
3: Construir a matriz  $\{\mathbf{D}_2^i\}_{i=1}^s$ ; //  $s$  : número de indivíduos da geração  $t$ 
4: para  $i \leftarrow 1$  até  $\text{length}(P(t))$  fazer
5:   Remover as colunas da matriz  $\mathbf{D}_2^i$  de acordo com os genes definidos para “zero”;
   // Poda de vetores-suporte
6:    $\mathbf{z}_2^i \leftarrow \mathbf{D}_2^i \mathbf{I}_2^i$ ; // Resolução com base na Pseudo-Inversa
7:   Avaliar a taxa de acerto do conjunto de treinamento; // Função de aptidão, Eq. 4.3
8: fim para
9: repetir
10:   $P_{new} \leftarrow P.\text{selecao}()$ ; // Seleciona as matrizes  $\mathbf{D}_2^i$ 
11:   $P_{new} \leftarrow P.\text{cruzamento}()$ ;
12:   $P_{new} \leftarrow P.\text{mutacao}()$ ;
13:   $t \leftarrow t + 1$ ;
14:  Criar uma matriz  $\mathbf{D}_2^i$  para cada indivíduo em  $P(t + 1)$ ;
15:  Remover as colunas da matriz  $\mathbf{D}_2^i$  de acordo com os genes definidos para “zero”;
   // Poda de vetores-suporte
16:   $P_{new}.\text{avaliacao}()$ ; // Avalia a taxa de acerto para cada  $\mathbf{D}_2^i$  pela Eq. 4.3. Função de
   aptidão
17:   $P \leftarrow P_{new}$ ;
18: até  $P_{new}.\text{criterioDeParada}(t_{max})$ 
19: return  $P_{new}.\text{melhorSolucao}()$ ; // i.e.,  $\mathbf{D}^o$ 

```

4.3.3 FSGAS-LSSVM (Proposta 2)

Esta abordagem tem por finalidade construir classificadores LSSVM com conjunto de vetores-suporte de tamanho fixo. O classificador *Fixed-Size Genetic Algorithm for Sparse LSSVM*, *FSGAS-LSSVM*, baseia-se em um AG com um único objetivo e codificação binária dos genes, em que os operadores de reprodução e mutação tradicionais devem ser modificados para que se mantenha a proporção previamente desejada com relação ao percentual de remoção desejado (SILVA; ROCHA NETO, 2015).

No classificador FSGAS-LSSVM, Algoritmos Genéticos são utilizados para selecionar um conjunto de tamanho fixo de genes que melhor represente o conjunto de vetores-suporte para o problema alvo. Isso é feito por meio da remoção de padrões não relevantes ou daqueles que podem ser corrompidos com ruído e que impedem o classificador de alcançar acurácia mais elevada. Vale ressaltar que o processo de treinamento do classificador proposto e descrito anteriormente (MOGAS-LSSVM), que também tem por base AGs, não gera conjuntos reduzidos de vetores-suporte de tamanho fixo.

4.3.3.1 Indivíduos (ou cromossomos)

Como dito anteriormente, as considerações descritas na Seção 4.3.2.1 também se aplicam à formulação, representação e forma de evolução dos indivíduos utilizados no classificador FSGAS-LSSVM.

Uma modificação é feita na iniciação dos genes que compõem cada indivíduo. No classificador MOGAS-LSSVM o indivíduo é iniciado de forma aleatória, o que não pode ser feito no classificador FSGAS-LSSVM devido a sua proposição ser voltada à construção de um classificador de tamanho fixo.

O que ocorre de fato é a criação de genes definidos com valores iguais a “zero” segundo um parâmetro que se encontra no intervalo $[0, 1]$ e que representa o percentual de vetores-suporte que se deseja remover. Por exemplo, suponha um problema cujos indivíduos possuem 250 genes e que deseja-se uma redução de 80% em termos de vetores-suporte, a iniciação dos genes se dará pelo seguinte fluxo:

1. Definir o parâmetro que representa a remoção de vetores-suporte para 0,8;
2. Criar um vetor de 0's de tamanho $0,8 * 250 = 200$;
3. Criar um vetor de 1's de tamanho $250 - 200 = 50$;
4. Concatenar os vetores criados nas etapas 2 e 3 em um único vetor v_{aux} ;
5. Embaralhar os elementos do vetor v_{aux} .

Dessa forma garante-se a criação de 50 genes definidos como “um”, que representam uma redução em 80% de vetores-suporte quando se considera a quantidade de padrões do referido problema. Posteriormente, o Algoritmo Genético irá determinar as melhores posições desses genes com o intuito de definir o conjunto de vetores-suporte mais adequado segundo a função de aptidão.

4.3.3.2 Reprodução

Como descrito na seção anterior, o principal objetivo do operador de reprodução nessa proposta é que o número de genes definidos com valor “um” seja constante. Isso é necessário devido à inadequação dos operadores de reprodução convencionais, uma vez que indivíduos inválidos são produzidos ao se utilizar tais operadores ³.

Assim, propõe-se o uso de uma codificação indireta dos indivíduos, para que seja possível transformar a codificação binária em uma codificação inteira correspondente e assim aplicar operadores de permutação como *Partially Matched Crossover (PMX)*, *Cyclic Crossover (CX)* ou *Order Crossover (OX)* (EIBEN; SMITH, 2003).

A Figura 18, modificada de Silva e Rocha Neto (2015), descreve o processo de transformação entre as codificações binárias e inteiras e, em seguida, a aplicação do operador de cruzamento OX1 sob a codificação inteira dos pais. Após este processo, cada descendente é transformado novamente para sua respectiva codificação binária de modo que cada filho esteja apto a compor a população seguinte.

O primeiro passo consiste em escolher de forma aleatória um subconjunto de genes que será mantido para a geração dos novos indivíduos. Na Figura 18, este subconjunto é definido pelos genes localizados nas posições 3, 4 e 5 de cada indivíduo *Pai*. Em seguida é realizada uma codificação indireta de cada indivíduo *Pai* por meio de uma nova representação através de um indivíduo de codificação inteira ⁴ de tal forma que a posição de cada gene de um indivíduo com codificação inteira está diretamente associado às mesmas posições dos genes de um indivíduo com codificação binária.

Os indivíduos codificados são então submetidos ao operador de reprodução OX1 que opera segundo discussão apresentada na Seção 3.2.2. Por fim, a conversão dos indivíduos obtidos após o processo de reprodução é realizada para a representação binária correspondente.

4.3.3.3 Mutação

Como já mencionado, o indivíduo do classificador FSGAS-LSSVM é representado por um vetor de genes binários em que cada gene é definido como “um” (verdadeiro), se um certo padrão é utilizado como vetor-suporte ou “zero” (falso) caso contrário. O operador de mutação utiliza um par de genes que trocam seu material genético.

³ Como normalmente é feito pelos operações de reprodução padrões: *um-ponto*, *dois-pontos* ou até mesmo *cruzamento uniforme* (SILVA; ROCHA NETO, 2015).

⁴ Isso se faz necessário devido o processo de manipulação de genes que ocorre em operadores de permutação não ser compatível com indivíduos representados por codificação binária.

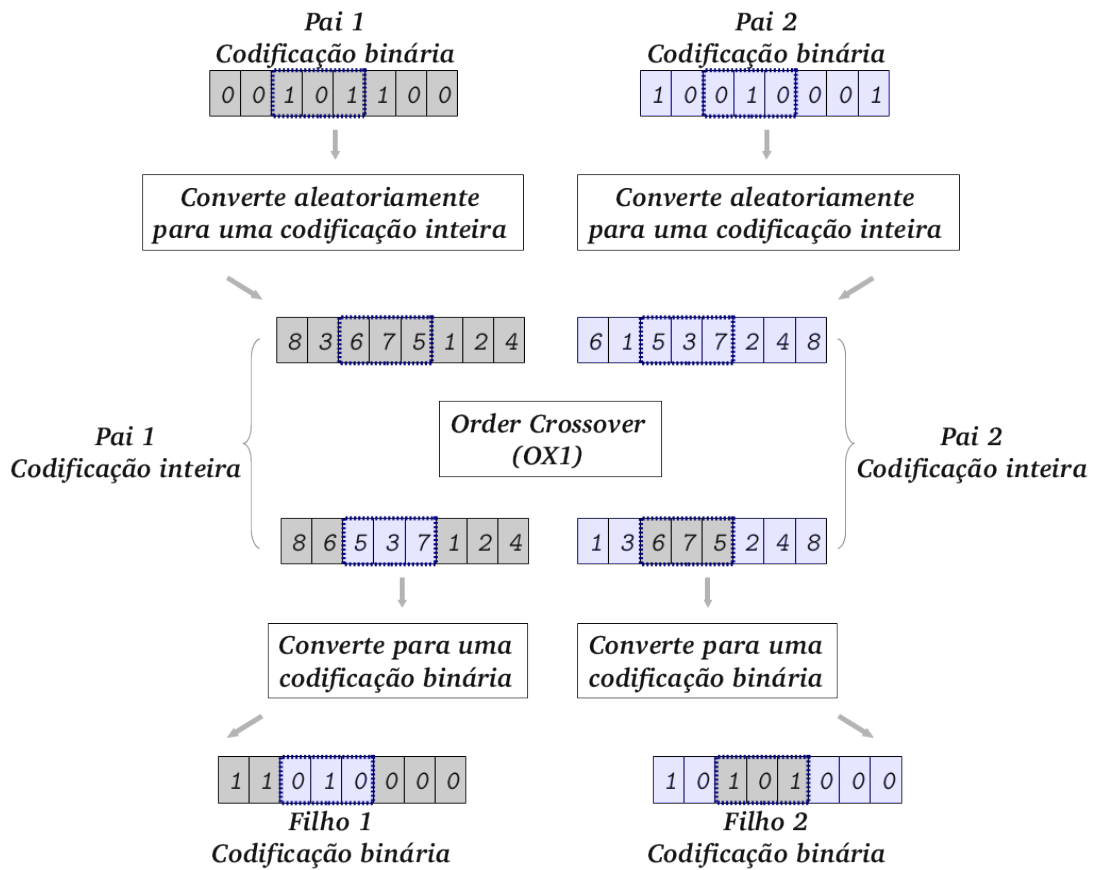


Figura 18: Processo de transformação entre codificação binária-inteira para aplicação de operador de cruzamento OX1 seguido pela transformação entre codificação inteira-binária.

A fim de manter o número de genes definido como “um” constante, o procedimento apresentado na Figura 19 é realizado. Inicialmente as posições dos genes 1’s e 0’s são identificadas. De acordo com um parâmetro que representa a probabilidade de mutação (10% utilizada nas simulações) e com base na quantidade de elementos definidos como “um”, calcula-se o número de mutações a serem realizadas. Em seguida dentre as posições de pares 1’s e 0’s previamente identificados, escolhem-se de forma aleatória, quais pares devem ter seu material genético trocado.

4.3.3.4 Função de Aptidão

A função de aptidão é representada por um problema de um único objetivo: a acurácia do classificador sobre o conjunto de treinamento (ou validação). A solução do sistema linear é obtida através do método da Pseudo-Inversa, uma vez que a matriz solução é não quadrada, resultante da poda de padrões que tenham valor do gene associado igual a zero.

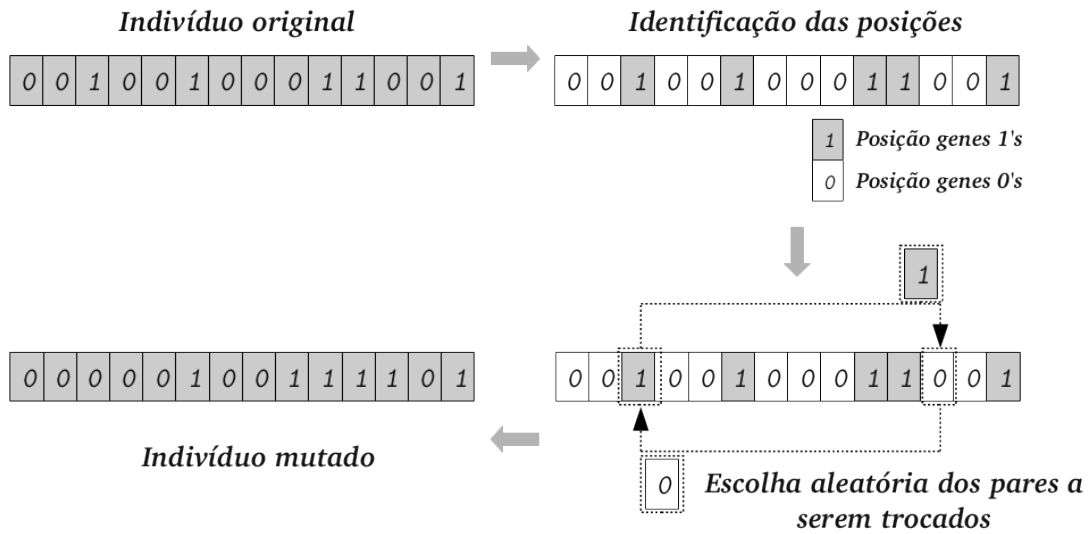


Figura 19: Esquema que descreve o procedimento de mutação onde garante-se a quantidade de genes definidos como “um” constante.

4.3.3.5 Parâmetros do Classificador FSGAS-LSSVM

Além dos parâmetros típicos do classificador LSSVM padrão μ , dos parâmetros de kernel e dos parâmetros relacionados ao processo de otimização do Algoritmo Genético, deve-se também informar previamente a quantidade de vetores-suporte que se deseja remover do classificador FSGAS-LSSVM.

4.3.3.6 Algoritmo FSGAS-LSSVM

O processo de treinamento do classificador FSGAS-LSSVM, em geral, segue o procedimento apresentado no Algoritmo 4. Contudo, vale ressaltar que o treinamento deste classificador apresenta particularidades no que se refere às modificações das funções de reprodução e mutação convencionais do AG para indivíduos binários e à necessidade em definir um valor para o parâmetro que estabelece o percentual de vetores-suporte a serem removidos.

Como já mencionado, com o objetivo de manter fixa a quantidade de genes definidos para “zero”, faz-se necessário a modificação da função de iniciação dos indivíduos, e dos operadores de cruzamento e mutação.

Dessa forma, o algoritmo que define o processo de treinamento do classificador FSGAS-LSSVM é apresentado no Algoritmo 5.

Algoritmo 5 Algoritmo do Classificador FSGAS-LSSVM

Entrada: t_{max} : número máximo de gerações, M : tamanho da população, percentual de vetores-suporte a serem removidos

Saída: melhor solução $\rightarrow \mathbf{D}^o$

- 1: $t \leftarrow 0$; // Inicia o contador de gerações
- 2: $P(t) \leftarrow \text{iniciaPopulacao}(M)$; // Neste caso um conjunto de genes com uma quantidade fixa de 0's
- 3: Construir a matriz $\{\mathbf{D}_2^i\}_{i=1}^s$; // s : número de indivíduos da geração t
- 4: **para** $i \leftarrow 1$ **até** $\text{length}(P(t))$ **fazer**
- 5: Remover as colunas da matriz $\mathbf{D}_2^i$ de acordo com os genes definidos para “zero”;
 // Poda de vetores-suporte
- 6: $\mathbf{z}_2^i \leftarrow \mathbf{D}_2^i \mathbf{1}_2$; // Resolução com base na Pseudo-Inversa
- 7: Avaliar a taxa de acerto do conjunto de treinamento; // Função de aptidão
- 8: **fim para**
- 9: **repetir**
- 10: $P_{new} \leftarrow P.\text{selecao}()$; // Seleciona as matrizes $\mathbf{D}_2^i$
- 11: $P_{new} \leftarrow P.\text{cruzamento}()$; // Operador OX1 descrito na Seção 4.3.3.2
- 12: $P_{new} \leftarrow P.\text{mutacao}()$; // Conforme descrito na Seção 4.3.3.3
- 13: $t \leftarrow t + 1$;
- 14: Criar uma matriz $\mathbf{D}_2^i$ para cada individuo em $P(t + 1)$;
- 15: Remover as colunas da matriz $\mathbf{D}_2^i$ de acordo com os genes definidos para “zero”;
 // Poda de vetores-suporte
- 16: $P_{new}.\text{avaliacao}()$; // Avalia a taxa de acerto para cada $\mathbf{D}_2^i$. Função de aptidão
- 17: $P \leftarrow P_{new}$;
- 18: **até** $P_{new}.\text{criterioDeParada}(t_{max})$
- 19: **return** $P_{new}.\text{melhorSolucao}()$; // i.e., $\mathbf{D}^o$

4.4 Conclusão

Conforme já mencionado, para lidar com a falta de esparsidade em classificadores LSSVM e obter conjuntos reduzidos em SVMs, várias soluções baseadas em conjunto reduzido e métodos de poda foram propostos. Tais técnicas visam simplificar a estrutura interna do modelo sem grandes impactos na superfície de decisão, uma vez que se deseja uma superfície o mais semelhante possível da original.

Alguns desses métodos e técnicas foram apresentadas neste capítulo, dentre elas dois novos classificadores foram propostos: MOGAS-LSSVM e FSGAS-LSSVM. Ambos visam a obtenção de conjuntos reduzidos em termos de vetores-suporte com base em Algoritmos Genéticos aplicados as Máquinas de Vetores-Suporte por Mínimos Quadrados.

Enquanto o classificador MOGAS-LSSVM destaca-se pelo uso de uma função de aptidão com base em escalarização que permite uma única função escalar representar os múltiplos objetivos a serem resolvidos, o classificador FSGAS-LSSVM é caracterizado por gerar um

conjunto reduzido de vetores-suporte de tamanho fixo em que o percentual de poda é definido *a priori*. Vale ressaltar que a função de aptidão do classificador MOGAS-LSSVM é ponderada por um custo de poda θ tal que o processo de otimização pode ser guiado somente em função da acurácia ($\theta = 1$) como em função da redução de vetores-suporte ($\theta = 0$).

Foi também apresentado e validado um novo procedimento de reprodução que visa manter fixa a quantidade de elementos em um indivíduo com codificação binária. Este procedimento contempla uma conversão no tipo de representação do indivíduo para que operadores de permutação sejam utilizados na geração dos novos indivíduos.

SIMULAÇÕES COMPUTACIONAIS

Neste capítulo é apresentada toda metodologia utilizada nas etapas de treinamento, validação e teste dos classificadores utilizados. As características das bases de dados escolhidas para verificação do desempenho dos classificadores propostos no Capítulo 4, bem como os resultados das simulações computacionais para classificadores clássicos e classificadores frutos desta dissertação também são expostas ao longo deste capítulo.

5.1 Metodologia

Para fins de avaliação do desempenho de generalização dos classificadores propostos, foram considerados conjuntos de dados de problemas do mundo real disponíveis no repositório de dados da *UCI Machine Learning Repository*. Foram utilizados quatro conjuntos de dados da UCI, a saber: *Diabetes*, *Haberman*, *Breast Cancer* e *Vertebral Column Pathologies* (LICHMAN, 2013).

Este último, originalmente de 3 classes, foi transformado em um problema binário de classificação através da junção das classes de patologias *hérnia de disco* e *espondilolistese* em apenas uma. Os demais problemas são originalmente de 2 classes. Além dos conjuntos citados anteriormente, os classificadores propostos foram ainda submetidos a avaliação por outros dois problemas conhecidos na literatura a saber: *Banana* e *Ripley*¹. A Tabela 4 contém informações relevantes sobre cada conjunto de dados utilizado, dentre elas: o número de padrões e de atributos.

Todos os classificadores utilizados foram implementados utilizando o ambiente *Matlab* e sua respectiva linguagem de programação nativa. Independente do conjunto de dados utilizado, o primeiro procedimento a ser realizado é a normalização de dados em função da sua média e desvio padrão como segue:

$$\mathbf{x}_{i_{norm}} = \frac{\mathbf{x}_i - \alpha_{\mathbf{x}}}{\sigma_{\mathbf{x}}}, \quad (5.1)$$

¹ Bases de dados sintéticas de 2 classes.

em que $\mathbf{x}_i$ representa o i -ésimo padrão do conjunto de dados, $\bar{\mathbf{x}}$ a média de todo o conjunto $\{\mathbf{x}\}_{i=1}^N$, $\sigma_{\mathbf{x}}$ o desvio padrão também de todo o conjunto $\{\mathbf{x}\}_{i=1}^N$ e $\mathbf{x}_{i_{norm}}$ o padrão $\mathbf{x}_i$ normalizado.

Tabela 4: Características dos conjuntos de dados utilizados neste trabalho.

Problema	Abreviação	# Padrões	# Atributos
Vertebral Column Pathologies	VCP	310	6
Haberman	HAB	306	3
Breast Cancer Winconsin	BCW	683	9
Pima Indians Diabets	PID	768	8
Ripley	RIP	1250	2
Banana	BAN	1001	2

Duas abordagens distintas foram utilizadas nas simulações computacionais de acordo com o tipo de classificador utilizado. A primeira abordagem (*Abordagem I*) consiste na divisão dos dados em treinamento e teste de tal maneira que 80% dos padrões são selecionados aleatoriamente para treinamento e o restante para teste. Os padrões selecionados para treinamento são utilizados para obtenção do classificador. Para que este seja avaliado em termos de generalização e resolução do problema, é necessário que padrões ainda não apresentados sejam fornecidos, ou seja, os padrões do conjunto de teste.

Na segunda abordagem (*Abordagem II*), a base de dados é dividida em treinamento, validação e teste de tal maneira que 60% dos padrões são selecionados para treinamento, 20% para validação e os 20% restantes para teste. Os dados do conjunto de validação são utilizados para avaliar a taxa de acerto do classificador ainda durante o processo de treinamento, diferentemente do que ocorre com a *Abordagem I*, na qual padrões já utilizados pelo classificador (ou seja, padrões do conjunto de treinamento) também são utilizados para tal finalidade.

Este procedimento é repetido por 30 realizações (rodadas) e por fim uma média da taxa de acerto (acurácia) do classificador em questão é obtida. A cada realização, de forma aleatória, uma nova divisão do conjunto de treinamento/teste (ou treinamento/validação/teste) é feita.

De um modo em geral, para os classificadores que utilizam Algoritmos Genéticos, inicialmente cria-se aleatoriamente uma população de soluções viáveis. Cada solução é uma cadeia de valores binários, onde cada bit representa a presença (1) ou ausência (0) de um padrão no conjunto de vetores-suporte, o que caracteriza se o multiplicador de Lagrange relacionado com um determinado padrão é diferente de zero ou não.

Conforme já mencionado, da geração t para a próxima geração $t + 1$ leva-se em conta o fato de que 10% dos melhores indivíduos são selecionados devido a um esquema de seleção elitista,

80% dos novos indivíduos são gerados como resultado da aplicação do operador de cruzamento e, em seguida, os indivíduos restantes são obtidos por mutação. Este processo é repetido por 30 gerações até que o melhor indivíduo seja obtido.

Cada população é composta por 120 indivíduos, cujo tamanho (i.e., o número de genes de um indivíduo) depende diretamente do problema em questão. Vale ressaltar que cada indivíduo representa uma determinada configuração a ser avaliada com base na função de aptidão definida de acordo com o classificador.

Um resumo a cerca das principais características da metodologia abordada é apresentado na Tabela 5.

Tabela 5: Resumo da metodologia abordada.

Normalização dos dados	Eq. (5.1)
Abordagem I	Divisão da base em treinamento (80%) e teste (20%)
Abordagem II	Divisão da base em treinamento (60%), validação (20%) e teste (20%)
Quantidade de realizações	30
Geracões (AG)	30
População (AG)	120 indivíduos
Parâmetros intrínsecos ao classificador	Definidos por tentativa e erro para o classificador SVM convencional e em seguida mantidos para os classificadores esparsos (SILVA; ROCHA NETO, 2015; ROCHA NETO; BARRETO, 2013; SILVA; ROCHA NETO, 2013)

O kernel utilizado para os classificadores LSSVM padrão, MOGAS-LSSVM e FSGAS-LSSVM, é do tipo *linear*, uma vez que o hiperplano ótimo de separação é encontrado no espaço de características. Vale ressaltar que os parâmetros intrínsecos ao classificador LSSVM² foram obtidos a partir de Silva e Rocha Neto (2015), Rocha Neto e Barreto (2013), Silva e Rocha Neto (2013). Em geral, a ideia é a obtenção do parâmetro do classificador por qualquer método (tal como a validação cruzada, tentativa e erro, entre outros, ver Seção 2.5) e, em seguida, usar os parâmetros obtidos nos classificadores propostos.

Uma vez obtido o classificador e concluído o procedimento de resolução ao longo das 30 realizações, pode-se definir como métricas de desempenho (parâmetros de comparação) entre os algoritmos avaliados:

² Neste caso, apenas o parâmetro de regularização μ , devido o kernel *linear* não necessitar de nenhum outro parâmetro.

- taxa média de acerto para o conjunto de teste (acurácia) juntamente com seu desvio padrão;
- percentual de redução do número de vetores-suporte utilizado por cada classificador na resolução do problema.

Os valores obtidos que representam as melhores métricas de desempenho, são destacados em negrito nas tabelas de resultados.

A Figura 20 apresenta a metodologia utilizada nas simulações computacionais por meio de um diagrama de blocos.

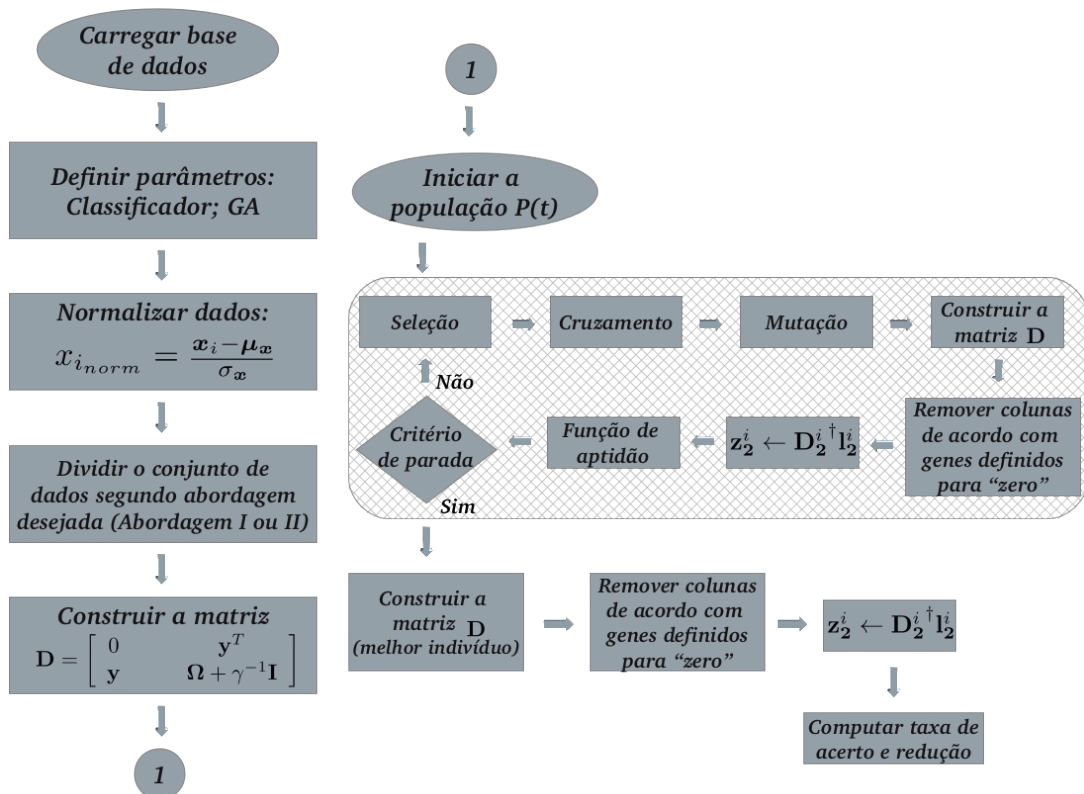


Figura 20: Diagrama de blocos da metodologia utilizada.

5.2 Resultados

Os resultados obtidos nas avalia  es dos classificadores MOGAS-LSSVM (Proposta 1) e FSGAS-LSSVM (Proposta 2) s o mostrados nesta sec  o. Para fins de compara  o s o apresentados ainda os resultados obtidos para classificadores cl ssicos.

5.2.1 Resultados para o Classificador MOGAS-LSSVM (Proposta 1)

Como descrito anteriormente, o classificador MOGAS-LSSVM visa tanto maximizar a acurácia do classificador como minimizar o número de vetores-suporte utilizados no processo de decisão.

Na Tabela 6, as métricas de desempenho obtidas utilizando a *Abordagem I* para 30 realizações independentes e com θ igual a zero são descritas. Mostra-se também o número médio de vetores-suporte (VS) ($\#VS_s$), o número de padrões de treinamento ($\#PT$), bem como os valores do parâmetro γ e a redução obtida (Red).

Tabela 6: Resultados para os classificadores LSSVM, IP-LSSVM, GAS-LSSVM e MOGAS-LSSVM com 80% (20%) do conjunto de dados para treinamento (teste).

Problema	Modelo	γ	Taxa de Acerto	# PT	# VS's	Red
VCP	LSSVM	0,05	$81,2 \pm 4,9$	248	248,0	–
	IP-LSSVM		$75,4 \pm 8,6$		184,0	25,8%
	GAS-LSSVM		$84,6 \pm 3,5$		116,5	53,0%
	MOGAS-LSSVM		$84,3 \pm 4,5$		53,0	78,6%
HAB	LSSVM	0,04	$73,8 \pm 5,0$	245	245,0	–
	IP-LSSVM		$70,6 \pm 4,9$		223,0	9,0%
	GAS-LSSVM		$76,1 \pm 4,5$		105,8	56,8%
	MOGAS-LSSVM		$75,7 \pm 4,6$		57,2	76,6%
BCW	LSSVM	0,04	$96,7 \pm 1,1$	546	546,0	–
	IP-LSSVM		$96,8 \pm 1,1$		284,0	48,0%
	GAS-LSSVM		$96,6 \pm 1,3$		273,1	49,9%
	MOGAS-LSSVM		$96,5 \pm 1,1$		146,8	73,1%
PID	LSSVM	0,04	$75,8 \pm 2,7$	614	614,0	–
	IP-LSSVM		$74,0 \pm 5,3$		503,0	18,1%
	GAS-LSSVM		$77,8 \pm 3,1$		280,4	54,3%
	MOGAS-LSSVM		$77,7 \pm 2,7$		199,6	67,5%
RIP	LSSVM	0,04	$87,7 \pm 1,6$	1000	1000,0	–
	IP-LSSVM		$86,4 \pm 3,3$		496,0	50,4%
	GAS-LSSVM		$87,7 \pm 2,2$		478,3	52,2%
	MOGAS-LSSVM		$88,1 \pm 2,7$		346,3	65,4%
BAN	LSSVM	0,04	$96,9 \pm 1,3$	801	801,0	–
	IP-LSSVM		$96,5 \pm 1,4$		394,0	50,8%
	GAS-LSSVM		$96,7 \pm 1,2$		394,2	50,7%
	MOGAS-LSSVM		$96,4 \pm 1,2$		245,5	69,4%

A fim de comparar o classificador MOGAS-LSSVM proposto neste trabalho com o classificador P-LSSVM e estar em conformidade com a sua metodologia de avaliação, foram realizadas simulações seguindo a *Abordagem II* (veja Seção 5.1).

Na Tabela 7 é possível verificar os resultados que relacionam os classificadores LSSVM, P-LSSVM, GAS-LSSVM e MOGAS-LSSVM.

Tabela 7: Resultados para os classificadores LSSVM, P-LSSVM, GAS-LSSVM e MOGAS-LSSVM com 60%, 20% e 20% do conjunto de dados para treinamento, validação e teste, respectivamente.

Problema	Modelo	γ	Taxa de Acerto	# PT	# VS's	Red
VCP	LSSVM	0,05	$80,1 \pm 3,2$	186	186,0	–
	P-LSSVM		$80,0 \pm 4,7$		120,5	35,2%
	GAS-LSSVM		$82,7 \pm 5,8$		92,6	50,1%
	MOGAS-LSSVM		$81,3 \pm 5,3$		29,4	84,2%
HAB	LSSVM	0,04	$73,7 \pm 3,6$	184	184,0	–
	P-LSSVM		$74,2 \pm 6,1$		94,0	48,9%
	GAS-LSSVM		$73,0 \pm 6,8$		95,7	47,9%
	MOGAS-LSSVM		$75,1 \pm 7,1$		23,1	87,5%
BCW	LSSVM	0,04	$96,9 \pm 0,8$	410	410,0	–
	P-LSSVM		$96,3 \pm 1,5$		210,3	48,7%
	GAS-LSSVM		$95,9 \pm 1,8$		206,2	49,7%
	MOGAS-LSSVM		$96,0 \pm 1,6$		91,0	77,8%
PID	LSSVM	0,04	$75,8 \pm 1,5$	461	461,0	–
	P-LSSVM		$75,6 \pm 3,3$		230,5	50,0%
	GAS-LSSVM		$77,0 \pm 3,5$		230,8	49,9%
	MOGAS-LSSVM		$76,9 \pm 3,9$		113,6	75,4%
RIP	LSSVM	0,04	$87,9 \pm 0,9$	750	750,0	–
	P-LSSVM		$87,8 \pm 1,7$		377,0	49,7%
	GAS-LSSVM		$87,6 \pm 2,1$		371,2	50,5%
	MOGAS-LSSVM		$87,9 \pm 2,1$		220,6	70,6%
BAN	LSSVM	0,04	$96,8 \pm 0,6$	601	601,0	–
	P-LSSVM		$97,1 \pm 0,9$		293,0	51,2%
	GAS-LSSVM		$97,0 \pm 1,2$		300,3	50,0%
	MOGAS-LSSVM		$97,0 \pm 1,3$		159,4	73,5%

Ao analisar os dados das Tabelas 6 e 7, pode-se concluir que o desempenho dos classificadores esparsos baseados em Algoritmos Genéticos (GAS-LSSVM e MOGAS-LSSVM) foram equivalentes aos alcançados pelos classificadores LSSVM, IP-LSSVM P-LSSVM. Em alguns casos, como mostrado na Tabela 6 para os problemas VCP, HAB e PID, os desempenhos dos classificadores esparsos baseados em AGs foram ainda melhor. Vale ressaltar que, como esperado, a proposta multiobjetivo alcança uma redução muito melhor em termos de vetores-suporte quando comparado com propostas de um único objetivo, uma vez que MOGAS-LSSVM incorpora a capacidade de reduzir o número de vetores-suporte ainda no processo de formação.

Nota-se que os valores de desvio padrão obtidos apresentam uma baixa variação. A exemplo, para o problema BAN na Tabela 6, tanto o classificador MOGAS-LSSVM quanto o classificador GAS-LSSVM, apresentam um valor médio de acurácia de 97,0% e valores de desvio de $\pm 1,3$ e $\pm 1,2$ respectivamente. Portanto, conclui-se que os valores de desvio padrão ratificam as acurácias apresentadas, uma vez que estão próximos.

Além dos resultados anteriores, na Figura 21a são apresentados os valores da taxa média de acerto para os classificadores LSSVM, GAS-LSSVM e MOGAS-LSSVM, enquanto na Figura 21b são mostrados os valores para a média de vetores-suporte para 10% até 80% do tamanho do conjunto de treinamento sobre 30 rodadas independentes para o conjunto de dados VCP. Vale ressaltar que o custo de poda utilizado para o classificador MOGAS-LSSVM foi $\theta = 1$.

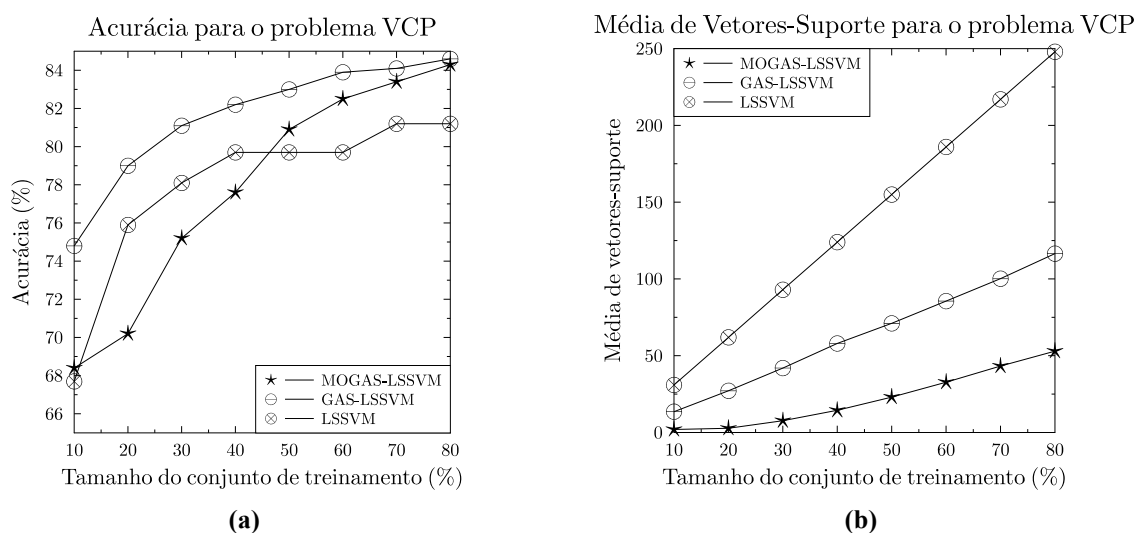


Figura 21: Comparação de desempenho entre os classificadores MOGAS-LSSVM, GAS-LSSVM e LSSVM para o problema VCP. (a) Acurácia *versus* variação do tamanho do conjunto de treinamento. (b) Média de VS *versus* variação do tamanho do conjunto de treinamento.

Nota-se na Figura 21a que as acurácias obtidas para o classificador GAS-LSSVM são maiores que as obtidas para o classificador LSSVM. Observa-se também que as acurácias obtidas para o classificador MOGAS-LSSVM proposto são maiores que as encontradas para o classificador LSSVM padrão quando o tamanho do conjunto de dados de treinamento é igual ou superior a 50%.

Além disso, a média de vetores-suporte para os modelos MOGAS-LSSVM e GAS-LSSVM são menores do que as verificadas para o classificador LSSVM. Isso enfatiza a importância do classificador proposto, pois mesmo com um baixo número de padrões, em um conjunto de dados de treinamento, a proposta prover um classificador com menor número de vetores-suporte e maior acurácia que o classificador LSSVM padrão.

A Figura 22, apresenta a acurácia e a média de vetores-suporte para diferentes custos de poda θ e diferentes tamanhos de conjuntos de dados de treinamento. À medida que o custo de poda aumenta, maior ênfase é dada a acurácia sobre o conjunto de teste, característica apresentada na Figura 22a. Pode-se ver também, na Figura 22b, que a média dos vetores-suporte depende diretamente do valor de θ . Nesta figura, por exemplo, para um conjunto de treinamento com 80% do tamanho da base de dados, verifica-se menos vetores-suporte com $\theta = 0$ em

comparação a $\theta = 1$. O mesmo pode ser dito para os outros tamanhos de conjunto de treinamento.

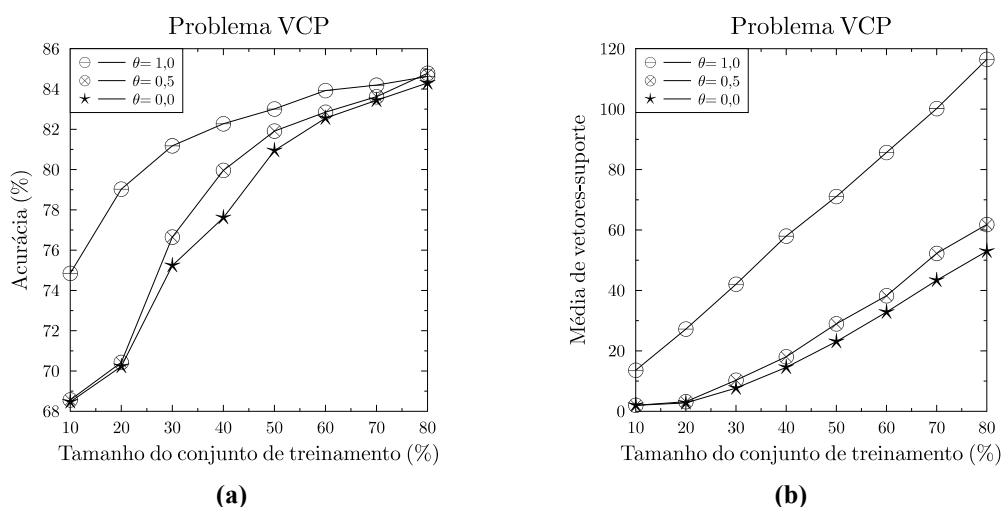


Figura 22: Desempenho do classificador MOGAS-LSSVM em função do custo de poda para o problema VCP: (a) Acurácia *versus* diferentes valores de tamanho do conjunto de treinamento em função do θ no intervalo de $[0, 1]$. (b) Média de vetores-suporte *versus* diferentes valores de tamanho do conjunto de treinamento em função do θ no intervalo de $[0, 1]$.

Na Figura 23 são apresentados os resultados para a acurácia e redução do conjunto de vetores-suporte em função do custo de poda θ e obtidos para cada problema avaliado segundo a *Abordagem I*.

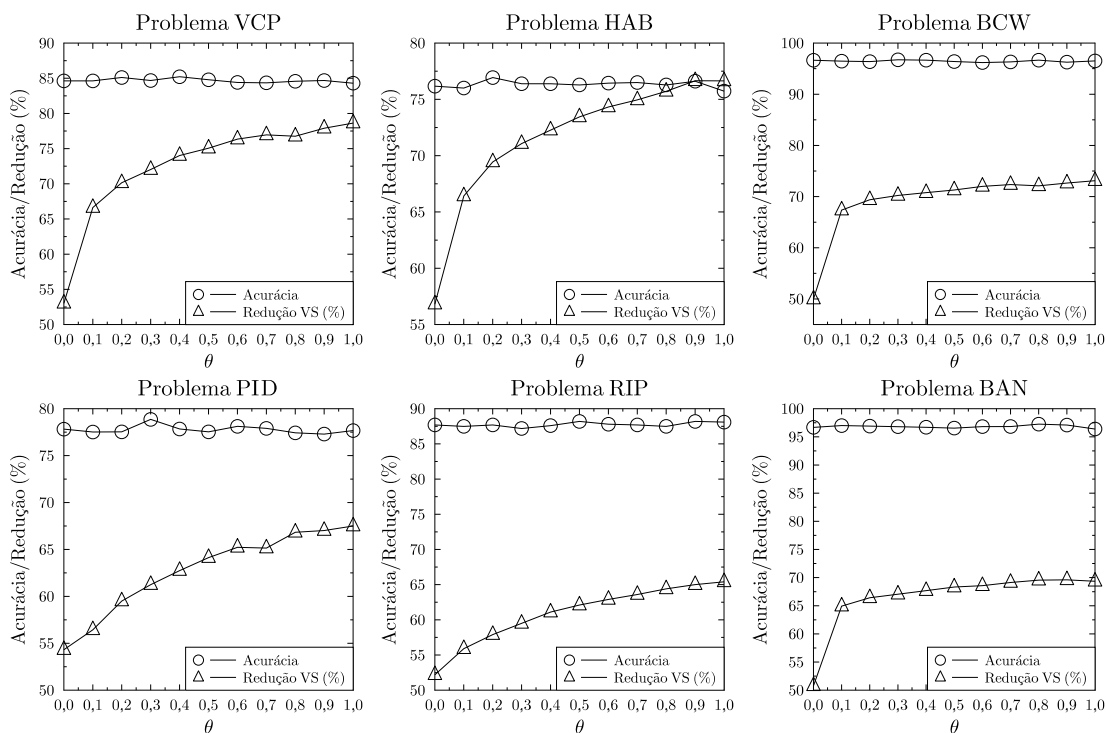


Figura 23: Acurácia e redução em função do custo de poda θ , obtidos para cada problema e avaliados com 80% e 20% do conjunto de treinamento e teste, respectivamente.

Na Figura 24, é igualmente apresentada a redução de vetores-suporte e acurácia em função do custo de poda θ obtidos para cada problema avaliado segundo a *Abordagem II*.

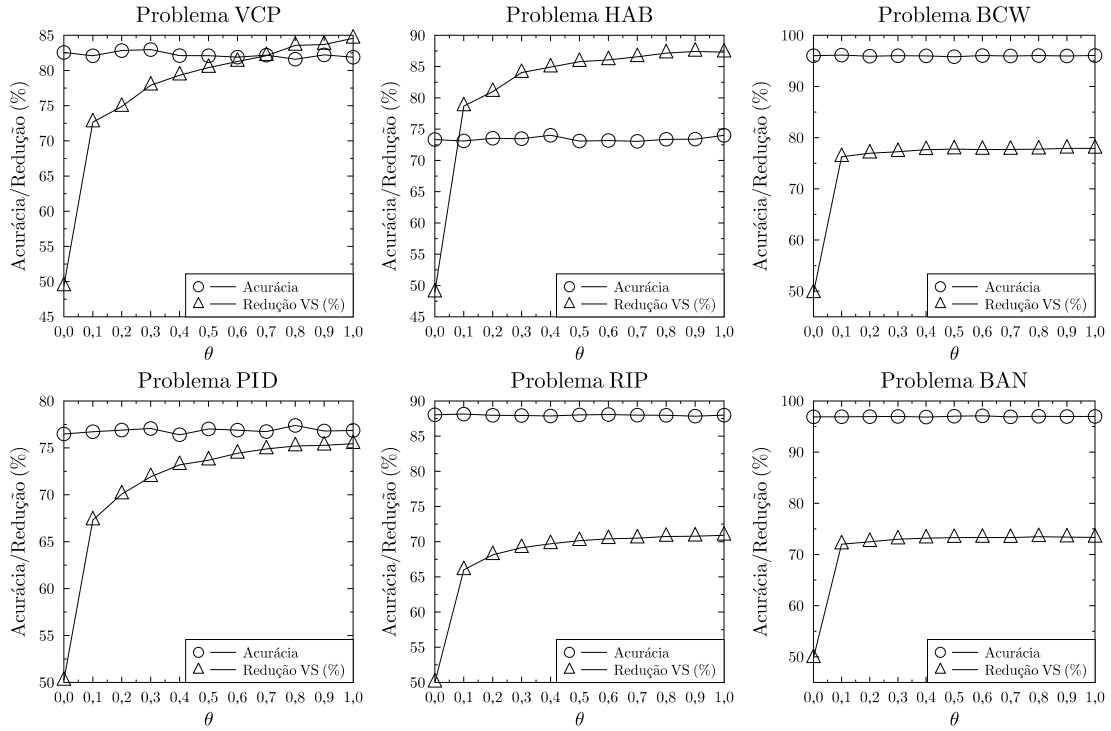


Figura 24: Acurácia e redução em função do custo de poda θ obtidos para cada problema avaliados com 60%, 20% e 20% do conjunto de treinamento, validação e teste, respectivamente.

Nas Figuras 23 e 24, pode-se notar que a acurácia se mantém quase constante em função da variação de θ dentro do intervalo de $[0,1]$ para cada conjunto de dados avaliado. Estes resultados mostram que o classificador MOGAS-LSSVM proposto nesta dissertação apresenta robustez em termos de acurácia.

Também é possível notar nessas figuras, a relação inversamente proporcional entre a redução de vetores-suporte e o valor de θ . Isto pode ser observado particularmente para os conjuntos de dados HAB, PID, RIP e VCP avaliados tanto na *Abordagem I* quanto na *Abordagem II*. Para os problemas BCW e BAN, o aumento na redução ocorre com menos intensidade ao longo da variação de θ dentro do intervalo $0.2 < \theta < 1$.

Em termos de tempo médio de treinamento do modelo proposto, foram realizadas medições para a maior e a menor base dentre os problemas apresentados na Tabela 4, sendo respectivamente as bases HAB e RIP. Tais resultados foram obtidos através do software matlab em um computador com processador intel core i5 (Intel coreTM i5-3470) e 8GB de RAM.

Tabela 8: Tempo médio de treinamento para os classificadores IP-LSSVM (*Abordagem I*), P-LSSVM (*Abordagem II*), LSSVM e MOGAS-LSSVM (*Abordagens I e II*).

Problema	Modelo		θ	Tempo Médio (seg.)	
				<i>Abordagem I</i>	<i>Abordagem II</i>
HAB	LSSVM		-	5,0E-3	4,8E-3
	IP-LSSVM ¹	P-LSSVM ²		5,8E-3	9,7E-3
	MOGAS-LSSVM		1	4,7E+0	4,3E+0
			0	5,4E+0	5,1E+0
RIP	LSSVM		-	92,5E-3	119,1E-3
	IP-LSSVM ¹	P-LSSVM ²		117,6E-3	155,8E-3
	MOGAS-LSSVM		1	81,8E+0	44,5E+0
			0	70,5E+0	57,0E+0

5.2.2 Resultados para o Classificador FSGAS-LSSVM (Proposta 2)

Tendo com objetivo a construção de classificadores LSSVM esparsos para tarefas de classificação de tamanho fixo em relação aos vetores-suporte para solução final, o classificador FSGAS-LSSVM baseia-se em um Algoritmo Genético com um único objetivo.

Através de um parâmetro fornecido inicialmente na execução do algoritmo, a quantidade de vetores-suporte a serem removidos permanece inalterada, garantindo um classificador de tamanho fixo em relação aos multiplicadores de Lagrange diferentes de 0. Dessa forma, a otimização realizada pelo uso de Algoritmos Genéticos é guiada pela busca das posições de vetores-suporte (genes do indivíduo cujo material genético é igual a 0) a serem removidos e que menos impacte na acurácia do classificador.

Na Tabela 9, as métricas de desempenho obtidas nas simulações computacionais são apresentadas utilizando a *Abordagem I*, bem como as características referentes ao número médio de VS ($\#SV^0s$), número de padrões utilizados no treinamento ($\#TP$), valores do parâmetro μ e por fim a redução desejada (Red). Os resultados considerados foram obtidos para os classificadores LSSVM, IP-LSSVM e FSGAS-LSSVM.

Já na Tabela 10, são mostrados os resultados obtidos para os classificadores LSSVM,

¹ A metodologia de avaliação do classificador IP-LSSVM é originalmente implementada segundo *Abordagem I*.

² A metodologia de avaliação do classificador P-LSSVM é originalmente implementada segundo *Abordagem II*.

P-LSSVM e FSGAS-LSSVM utilizando a *Abordagem II*. Como apresentada anteriormente, nesta abordagem 60% dos padrões são utilizados no conjunto de treinamento, 20% são utilizados no conjunto de validação e os 20% restante são utilizados nos testes.

Tabela 9: Resultados para os classificadores LSSVM, IP-LSSVM e FSGAS-LSSVM com 80% (20%) do conjunto de dados de treinamento (teste).

Problema	Modelo	γ	Taxa de Acerto	# PT	# VS's	Red
VCP	LSSVM	0.05	$81,2 \pm 4,9$	248	248,0	—
	IP-LSSVM		$75,1 \pm 6,6$		198,0	20,0%
			$54,1 \pm 17,8$		124,0	50,0%
			$36,4 \pm 7,7$		50,0	80,0%
	FSGAS-LSSVM		$83,7 \pm 3,7$		198,0	20,0%
			$83,6 \pm 4,1$		124,0	50,0%
			$81,2 \pm 3,9$		50,0	80,0%
HAB	LSSVM	0.04	$73,8 \pm 5,0$	245	245,0	—
	IP-LSSVM		$57,6 \pm 9,4$		196,0	20,0%
			$52,0 \pm 12,5$		122,0	50,0%
			$40,7 \pm 17,6$		49,0	80,0%
	FSGAS-LSSVM		$74,3 \pm 4,5$		196,0	20,0%
			$75,4 \pm 6,2$		122,0	50,0%
			$74,4 \pm 4,8$		49,0	80,0%
BCW	LSSVM	0.04	$96,7 \pm 1,1$	546	546,0	—
	IP-LSSVM		$96,9 \pm 1,1$		437,0	20,0%
			$96,9 \pm 1,3$		273,0	50,0%
			$97,4 \pm 1,0$		109,0	80,0%
	FSGAS-LSSVM		$95,9 \pm 1,6$		437,0	20,0%
			$95,9 \pm 1,8$		273,0	50,0%
			$95,9 \pm 1,6$		109,0	80,0%
PID	LSSVM	0.04	$75,8 \pm 1,1$	614	614,0	—
	IP-LSSVM		$71,6 \pm 8,4$		491,0	20,0%
			$30,6 \pm 4,6$		307,0	50,0%
			$28,9 \pm 4,0$		123,0	80,0%
	FSGAS-LSSVM		$77,5 \pm 3,7$		491,0	20,0%
			$76,4 \pm 3,3$		307,0	50,0%
			$76,6 \pm 3,1$		123,0	80,0%
RIP	LSSVM	0.04	$87,7 \pm 1,6$	1000	1000,0	—
	IP-LSSVM		$87,7 \pm 1,7$		800,0	20,0%
			$86,1 \pm 2,9$		500,0	50,0%
			$15,2 \pm 5,4$		200,0	80,0%
	FSGAS-LSSVM		$87,8 \pm 2,2$		800,0	20,0%
			$87,7 \pm 1,9$		500,0	50,0%
			$87,9 \pm 2,0$		200,0	80,0%
BAN	LSSVM	0.04	$96,9 \pm 1,3$	801	801,0	—
	IP-LSSVM		$96,7 \pm 1,2$		641,0	20,0%
			$96,7 \pm 1,3$		400,0	50,0%
			$90,3 \pm 5,8$		160,0	80,0%
	FSGAS-LSSVM		$96,8 \pm 1,0$		641,0	20,0%
			$96,9 \pm 1,0$		400,0	50,0%
			$96,6 \pm 0,9$		160,0	80,0%

Tabela 10: Resultados para os classificadores LSSVM, P-LSSVM e FSGAS-LSSVM com 60%, 20% e 20% do conjunto de dados para treinamento, validação e teste, respectivamente.

Problema	Modelo	γ	Taxa de Acerto	# PT	# VS's	Red
VCP	LSSVM	0,05	80,1 $\pm$ 3,2	186	186,0	–
	P-LSSVM		69,1 $\pm$ 14,1		149,0	20,0%
			54,7 $\pm$ 14,2		93,0	50,0%
			50,9 $\pm$ 14,9		37,0	80,0%
			81,7 $\pm$ 4,4		149,0	20,0%
			81,1 $\pm$ 4,8		93,0	50,0%
	FSGAS-LSSVM		81,1 $\pm$ 4,4		37,0	80,0%
HAB	LSSVM	0,04	73,7 $\pm$ 3,6	184	184,0	–
	P-LSSVM		64,6 $\pm$ 6,9		147,0	20,0%
			63,1 $\pm$ 8,4		92,0	50,0%
			70,9 $\pm$ 6,7		37,0	80,0%
			72,5 $\pm$ 5,8		147,0	20,0%
			75,0 $\pm$ 7,4		92,0	50,0%
	FSGAS-LSSVM		73,7 $\pm$ 5,0		37,0	80,0%
BCW	LSSVM	0,04	96,9 $\pm$ 0,8	410	410,0	–
	P-LSSVM		96,2 $\pm$ 1,5		328,0	20,0%
			96,4 $\pm$ 1,6		205,0	50,0%
			95,6 $\pm$ 5,3		82,0	80,0%
			96,4 $\pm$ 1,6		328,0	20,0%
			96,1 $\pm$ 1,8		205,0	50,0%
	FSGAS-LSSVM		96,1 $\pm$ 1,0		82,0	80,0%
PID	LSSVM	0,04	75,8 $\pm$ 1,5	461	461,0	–
	P-LSSVM		68,9 $\pm$ 9,8		369,0	20,0%
			33,3 $\pm$ 3,3		230,0	50,0%
			30,2 $\pm$ 3,3		92,0	80,0%
			76,8 $\pm$ 2,4		369,0	20,0%
			77,1 $\pm$ 3,0		230,0	50,0%
	FSGAS-LSSVM		76,9 $\pm$ 3,0		92,0	80,0%
RIP	LSSVM	0,04	87,9 $\pm$ 0,9	750	750,0	–
	P-LSSVM		87,8 $\pm$ 2,0		600,0	20,0%
			83,3 $\pm$ 8,3		375,0	50,0%
			16,7 $\pm$ 8,0		150,0	80,0%
			88,2 $\pm$ 1,5		600,0	20,0%
			87,9 $\pm$ 1,9		375,0	50,0%
	FSGAS-LSSVM		87,7 $\pm$ 1,7		150,0	80,0%
BAN	LSSVM	0,04	96,8 $\pm$ 0,6	601	601,0	–
	P-LSSVM		96,8 $\pm$ 1,2		481,0	20,0%
			96,4 $\pm$ 1,5		300,0	50,0%
			89,6 $\pm$ 6,1		120,0	80,0%
			97,1 $\pm$ 0,9		481,0	20,0%
			96,8 $\pm$ 1,2		300,0	50,0%
	FSGAS-LSSVM		97,1 $\pm$ 0,8		120,0	80,0%

Para fins de comparação justa do classificador proposto FSGAS-LSSVM, o modelo P-LSSVM teve seu critério de parada alterado, que originalmente consiste na diminuição do desempenho do conjunto de validação. A modificação realizada consiste em adotar como critério de parada a redução desejada (valor passado como parâmetro para o classificador FSGAS-LSSVM) independente da acurácia verificada sob o conjunto de validação. Assim,

compara-se os desempenhos obtidos com a mesma quantidade de vetores-suporte utilizados para ambos os classificadores.

Com base nos resultados mostrados nas Tabelas 9 e 10, verifica-se que o desempenho do classificador esparsos de tamanho fixo (FSGAS-LSSVM) é equivalente ou mesmo superior, aos conseguidos pelos classificadores utilizados para comparação, tanto na *Abordagem I* (IP-LSSVM) como na *Abordagem II* (P-LSSVM).

A análise feita acerca dos valores de desvio padrão na Secção 5.2.1, também pode ser verificada nas Tabelas 9 e 10. A exemplo, para o caso do problema PID na Tabela 9, observam-se valores de acurácia em $87,7\% \pm 1,7$ e $87,9\% \pm 0,9$, respectivamente para os classificadores FSGAS-LSSVM e LSSVM.

A Figura 25a apresenta os valores de acurácia para diferentes tamanhos de conjuntos de dados de treinamento para o problema VCP. Como se espera, a medida que o percentual de poda previamente definido aumenta, o valor da acurácia diminui. Isto justifica-se devido a função de aptidão ser mono-objetivo (somente em função da acurácia). Já na Figura 25b são apresentados os valores de vetores-suporte utilizados também em função dos diferentes tamanhos de conjuntos de dados de treinamento para o problema VCP.

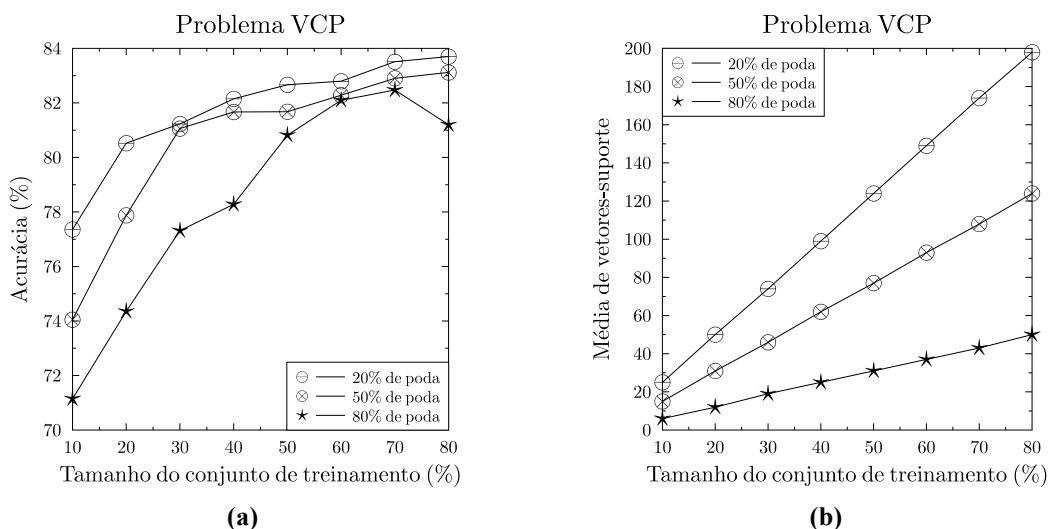


Figura 25: Desempenho do classificador FSGAS-LSSVM para o problema VCP em função do percentual de poda de vetores-suporte previamente definido: (a) Acurácia versus variação do tamanho do conjunto de treinamento. (b) Média de vetores-suporte versus variação do tamanho do conjunto de treinamento.

As Figuras 26 e 27 ilustram os valores de acurácia para 10% até 90% de remoção dos vetores-suporte, respectivamente para os classificadores FSGAS-LSSVM versus IP-LSSVM e FSGAS-LSSVM versus P-LSSVM.

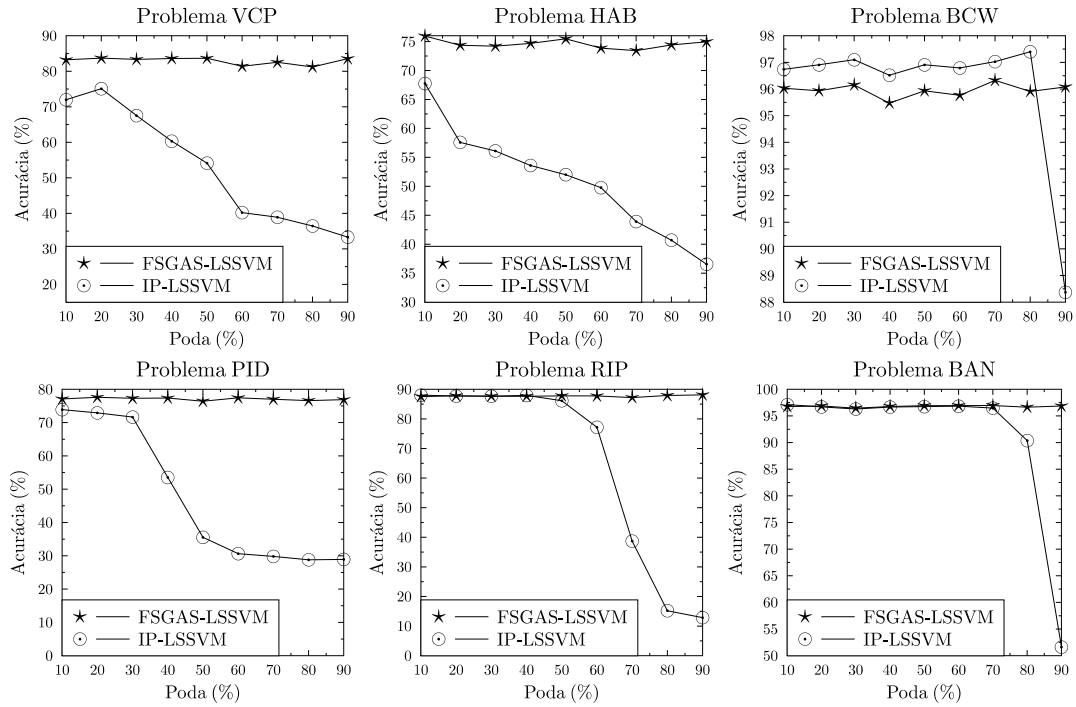


Figura 26: Comparação entre acurácia dos métodos FSGAS-LSSVM e IP-LSSVM em função do percentual de remoção de vetores-suporte para cada problema avaliado (*Abordagem I*).

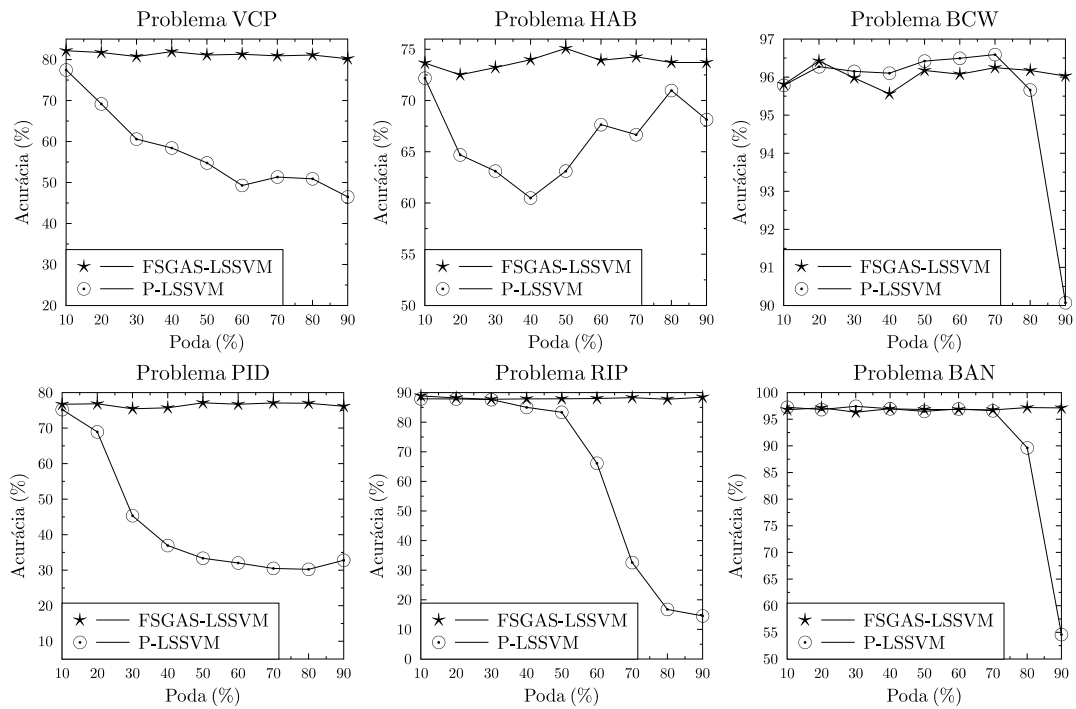


Figura 27: Comparação entre acurácia dos métodos FSGAS-LSSVM e P-LSSVM em função do percentual de remoção de vetores-suporte para cada problema avaliado (*Abordagem II*).

É possível notar que os valores obtidos para o classificador FSGAS-LSSVM são superiores aos resultados encontrados pelos modelos IP-LSSVM e P-LSSVM para cada percentual de remoção de vetores-suporte, exceto, para o conjunto de dados BCW. Particularmente para o problema BCW, os classificadores FSGAS-LSSVM, IP-LSSVM e P-LSSVM apresentam acurácias equiparáveis até cerca de 83% de redução de vetores-suporte, acima desse valor verifica-se uma queda de desempenho para os classificadores IP-LSSVM e P-LSSVM. Vale ressaltar que tanto para a *Abordagem I* (Figura 26) quanto para a *Abordagem II* (Figura 27), o classificador FSGAS-LSSVM mantém seu desempenho mesmo para percentuais de poda acima de 83%.

Similar ao apresentado para o classificador MOGAS-LSSVM, a Tabela 11 apresenta os tempos médios de treinamento para o modelo proposto FSGAS-LSSVM também obtidos através do software matlab em um computador com processador intel core i5 (Intel coreTM i5-3470) e 8GB de RAM.

Tabela 11: Tempo médio de treinamento para os classificadores IP-LSSVM (*Abordagem I*), P-LSSVM (*Abordagem II*), LSSVM e FSGAS-LSSVM (*Abordagens I e II*).

Problema	Modelo		Red.	Tempo Médio (seg.)	
				Abordagem I	Abordagem II
HAB	LSSVM		-	5,0E-3	4,8E-3
	IP-LSSVM ¹	P-LSSVM ²	20%	7,8E-3	5,9E-3
			50%	5,8E-3	9,7E-3
			80%	4,3E-3	12,2E-3
	FSGAS-LSSVM		20%	11,4E+0	7,7E+0
			50%	9,1E+0	6,3E+0
			80%	7,1E+0	4,8E+0
	RIP	LSSVM		-	92,5E-3
IP-LSSVM ¹		P-LSSVM ²	20%	149,7E-3	88,2E-3
			50%	117,6E-3	155,8E-3
			80%	105,9E-3	244,7E-3
FSGAS-LSSVM		20%	232,0E+0	105,0E+0	
		50%	131,7E+0	61,6E+0	
		80%	59,6E+0	36,3E+0	

¹ A metodologia de avaliação do classificador IP-LSSVM é originalmente implementada segundo *Abordagem I*.

² A metodologia de avaliação do classificador P-LSSVM é originalmente implementada segundo *Abordagem II*.

5.2.3 Comparação entre os classificadores

Os resultados apresentados nesta secção relacionam os valores de acurácia obtidos pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de dados utilizado no processo de treinamento para cada problema analisado.

A escolha dos valores utilizados para o parâmetro que representa o custo de poda (θ) do classificador MOGAS-LSSVM é feita a partir da melhor acurácia correspondente, verificada respectivamente para cada tamanho do conjunto de treinamento avaliado. São apresentadas ainda as métricas de desempenho do conjunto de teste para 30 realizações independentes.

Como o objetivo principal dessa comparação consiste em avaliar a acurácia dos classificadores para diferentes tamanhos do conjunto de dados utilizados para treinamento, o classificador P-LSSVM não satisfaz às condições necessárias de simulação. Originalmente, a base de dados utilizada em um classificador P-LSSVM é dividida segundo *Abordagem II*, pois o critério de parada do modelo ocorre em função da diminuição de desempenho sob o conjunto de validação. Por essa razão, o classificador P-LSSVM não será levado em consideração nas comparações.

1. Comparações para o problema VCP

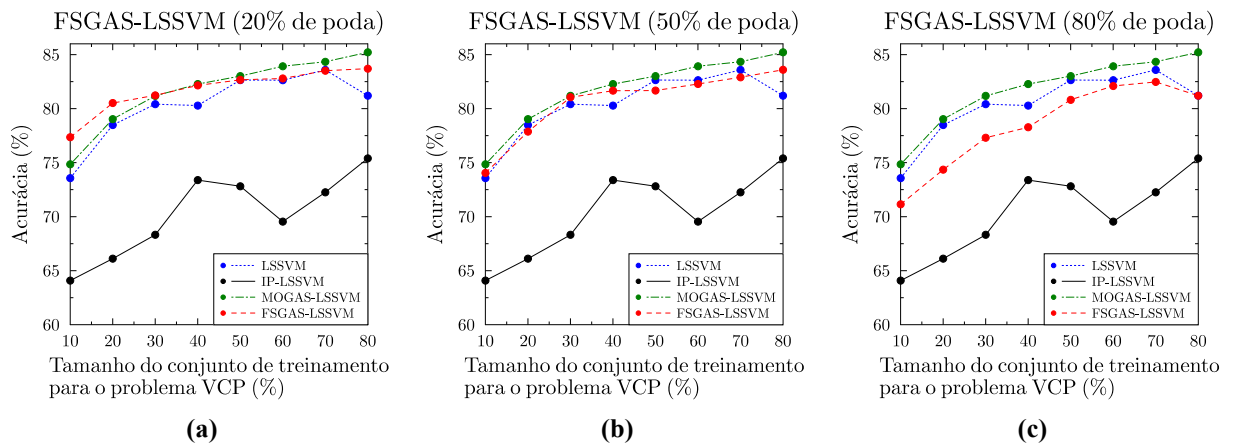


Figura 28: Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema VCP. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.

A partir da Figura 28a é possível verificar que até cerca de 30% do conjunto de dados para treinamento o classificador FSGAS-LSSVM obtém um desempenho superior aos demais modelos. Já o classificador MOGAS-LSSVM, apresenta-se superior para valores acima de 40%.

Particularmente para um conjunto com 30% e 40% dos dados para treinamento, verifica-se que os classificadores MOGAS-LSSVM e FSGAS-LSSVM são equiparáveis. Já para valores de dados de treinamento entre 50% e 70%, os classificadores FSGAS-LSSVM e LSSVM apresentam valores de acurácia equiparáveis. Nota-se ainda que o método IP-LSSVM apresenta-se inferior aos demais classificadores.

Conforme apresentado na Figura 28b, observa-se que o classificador MOGAS-LSSVM é superior aos demais modelos. Também é verificado que, mesmo com um percentual de poda definido para 50%, o classificador FSGAS-LSSVM apresenta valores de acurácias próximos aos obtidos pelo classificador LSSVM padrão, exceto para um conjunto de treinamento em 80%, onde o método proposto (FSGAS-LSSVM) supera o classificador convencional (LSSVM). Similarmente ao verificado na Figura 28a, o método IP-LSSVM apresenta-se inferior quando comparado aos demais classificadores.

Na Figura 28c, novamente o classificador MOGAS-LSSVM apresenta-se superior aos demais. O classificador FSGAS-LSSVM apresenta valores de acurácia superior ao método IP-LSSVM e inferior aos classificadores MOGAS-LSSVM e LSSVM padrão (exceto para um conjunto com 80% dos dados para treinamento).

Tal fato pode ser justificado devido a quantidade de vetores-suporte utilizados por cada classificador. A exemplo, considerando 10% do conjunto de dados para treinamento, o classificador FSGAS-LSSVM utiliza apenas 6 padrões de treinamento como vetores-suporte (conforme apresentado na Tabela 12).

A Tabela 12 apresenta as métricas de desempenho do conjunto de teste considerando 10% até 80% do conjunto de dados utilizados para treinamento ($\#DT$), o número médio de vetores-suporte (VS) ($\#VS$), o número de padrões de treinamento ($\#PT$), bem como os valores do custo de poda θ (MOGAS-LSSVM) e a redução obtida (Red) (ou definida, para o caso do classificador FSGAS-LSSVM).

Tabela 12: Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% até 80% do conjunto de dados para treinamento considerando o problema VCP ($\nu = 0,05$).

Modelo	θ	Taxa de Acerto	# PT	# VS's	Red	# DT.
LSSVM	-	$73,5 \pm 4,3$	31	31,0	–	10%
IP-LSSVM		$64,0 \pm 0,1$		23,0	25,8%	
MOGAS-LSSVM	0,0	$74,8 \pm 3,4$		14,0	54,8%	
FSGAS-LSSVM	-	$77,3 \pm 3,4$		25,0	20,0%	
		$74,0 \pm 4,9$		15,0	50,0%	
		$71,1 \pm 3,5$		6,0	80,0%	
LSSVM	-	$78,4 \pm 4,9$	62	62,0	–	20%
IP-LSSVM		$66,1 \pm 0,1$		46,0	25,8%	
MOGAS-LSSVM	0,0	$79,0 \pm 2,3$		27,0	56,4%	
FSGAS-LSSVM	-	$80,5 \pm 2,8$		50,0	20,0%	
		$77,8 \pm 2,7$		31,0	50,0%	
		$74,3 \pm 3,2$		12,0	80,0%	
LSSVM	-	$80,4 \pm 3,1$	93	93,0	–	30%
IP-LSSVM		$68,3 \pm 0,1$		69,0	25,8%	
MOGAS-LSSVM	0,0	$81,1 \pm 3,3$		42,0	54,8%	
FSGAS-LSSVM	-	$81,2 \pm 2,4$		74,0	20,0%	
		$81,0 \pm 2,6$		46,0	50,0%	
		$77,3 \pm 3,4$		19,0	80,0%	
LSSVM	-	$80,2 \pm 3,7$	124	124,0	–	40%
IP-LSSVM		$73,3 \pm 0,0$		92,0	25,8%	
MOGAS-LSSVM	0,0	$82,2 \pm 3,1$		58,0	53,2%	
FSGAS-LSSVM	-	$82,1 \pm 2,8$		99,0	20,0%	
		$81,6 \pm 2,9$		62,0	50,0%	
		$78,2 \pm 3,6$		25,0	80,0%	
LSSVM	-	$82,6 \pm 3,5$	155	155,0	–	50%
IP-LSSVM		$72,8 \pm 0,1$		115,0	25,8%	
MOGAS-LSSVM	0,0	$83,0 \pm 3,3$		71,0	54,1%	
FSGAS-LSSVM	-	$82,6 \pm 3,2$		124,0	20,0%	
		$81,6 \pm 3,1$		77,0	50,0%	
		$80,8 \pm 3,1$		31,0	80,0%	
LSSVM	-	$82,6 \pm 3,1$	186	186,0	–	60%
IP-LSSVM		$69,5 \pm 0,1$		138,0	25,8%	
MOGAS-LSSVM	0,0	$83,9 \pm 3,2$		86,0	53,7%	
FSGAS-LSSVM	-	$82,7 \pm 3,0$		149,0	20,0%	
		$82,2 \pm 3,4$		93,0	50,0%	
		$82,0 \pm 4,1$		37,0	80,0%	
LSSVM	-	$83,5 \pm 3,2$	217	217,0	–	70%
IP-LSSVM		$72,2 \pm 0,1$		161,0	25,8%	
MOGAS-LSSVM	0,7	$84,3 \pm 2,5$		48,0	77,8%	
FSGAS-LSSVM	-	$82,4 \pm 4,0$		43,0	20,0%	
		$82,9 \pm 3,0$		108,0	50,0%	
		$83,5 \pm 3,8$		174,0	80,0%	
LSSVM	-	$81,2 \pm 4,9$	248	248,0	–	80%
IP-LSSVM		$75,4 \pm 8,6$		184,0	25,8%	
MOGAS-LSSVM	0,4	$85,2 \pm 4,4$		64,0	74,1%	
FSGAS-LSSVM	-	$83,7 \pm 3,7$		198,0	20,0%	
		$83,6 \pm 4,1$		124,0	50,0%	
		$81,2 \pm 3,9$		50,0	80,0%	

2. Comparações para o problema HAB

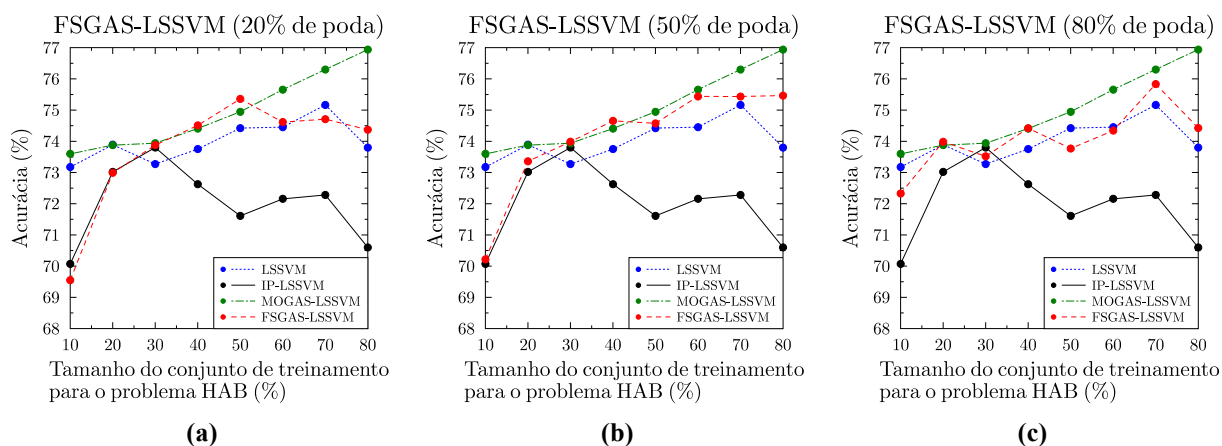


Figura 29: Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema HAB. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.

Na Figura 29a é possível observar que entre 20% e 50% dos dados para treinamento os classificadores avaliados (exceto IP-LSSVM) possuem valores de acurácia próximos, e para os demais valores, pode-se dizer que o classificador MOGAS-LSSVM é superior. Entre 10% e 30% dos dados para treinamento, os classificadores FSGAS-LSSVM e IP-LSSVM apresentam um comportamento similar caracterizado pela melhora do desempenho de aproximadamente 70% para 74%.

A Figura 29b apresenta uma maior proximidade em termos de acurácia entre os classificadores LSSVM, MOGAS-LSSVM e FSGAS-LSSVM (exceto para 10% dos dados para treinamento). É possível verificar uma melhora gradativa da acurácia para o classificador MOGAS-LSSVM a partir de 50% dos dados para treinamento. Assim como na Figura 29a, o classificador IP-LSSVM apresenta um desempenho inferior aos demais modelos a partir de 35% dos dados para treinamento.

Já na Figura 29c, de um modo em geral, os classificadores LSSVM e FSGAS-LSSVM são equiparáveis. O classificador MOGAS-LSSVM novamente apresenta-se superior aos demais classificadores e a proposta FSGAS-LSSVM alcança valores de acurácia próximo aos valores obtidos pelo modelo MOGAS-LSSVM entre 20% e 70% do conjunto de dados para treinamento.

A Tabela 13 apresenta as métricas de desempenho do conjunto de teste.

Tabela 13: Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% até 80% do conjunto de dados para treinamento considerando o problema HAB ($\nu = 0,04$).

Modelo	θ	Taxa de Acerto	# PT	# VS's	Red	# DT.
LSSVM	-	$73,1 \pm 2,1$	31	31,0	–	10%
IP-LSSVM		$70,0 \pm 0,1$		29,0	6,4%	
MOGAS-LSSVM	1,0	$73,6 \pm 1,3$		2,0	93,5%	
FSGAS-LSSVM	-	$69,5 \pm 7,2$		25,0	20,0%	
		$70,2 \pm 7,2$		15,0	50,0%	
		$72,3 \pm 4,2$		6,0	80,0%	
LSSVM	-	$73,8 \pm 1,7$	61	61,0	–	20%
IP-LSSVM		$73,0 \pm 0,0$		57,0	6,5%	
MOGAS-LSSVM	0,6	$73,8 \pm 1,2$		2,0	96,6%	
FSGAS-LSSVM	-	$72,9 \pm 2,4$		49,0	20,0%	
		$73,3 \pm 1,7$		30,0	50,0%	
		$73,9 \pm 1,5$		12,0	80,0%	
LSSVM	-	$73,2 \pm 2,0$	92	92,0	–	30%
IP-LSSVM		$73,8 \pm 0,0$		86,0	6,5%	
MOGAS-LSSVM	0,6	$73,9 \pm 2,3$		6,0	93,6%	
FSGAS-LSSVM	-	$73,8 \pm 1,6$		74,0	20,0%	
		$73,9 \pm 2,1$		46,0	50,0%	
		$73,5 \pm 1,8$		18,0	80,0%	
LSSVM	-	$73,7 \pm 2,1$	122	122,0	–	40%
IP-LSSVM		$72,6 \pm 0,0$		115,0	5,7%	
MOGAS-LSSVM	0,0	$74,4 \pm 2,1$		52,0	57,2%	
FSGAS-LSSVM	-	$74,5 \pm 2,0$		98,0	20,0%	
		$74,6 \pm 2,9$		61,0	50,0%	
		$74,4 \pm 2,6$		24,0	80,0%	
LSSVM	-	$74,4 \pm 2,8$	153	153,0	–	50%
IP-LSSVM		$71,6 \pm 0,0$		144,0	5,8%	
MOGAS-LSSVM	0,8	$74,9 \pm 2,4$		21,0	86,2%	
FSGAS-LSSVM	-	$75,3 \pm 2,5$		122,0	20,0%	
		$74,5 \pm 2,0$		76,0	50,0%	
		$73,7 \pm 2,8$		31,0	80,0%	
LSSVM	-	$74,4 \pm 3,3$	184	184,0	–	60%
IP-LSSVM		$72,1 \pm 0,0$		173,0	5,9%	
MOGAS-LSSVM	0,8	$75,6 \pm 2,7$		30,0	83,6%	
FSGAS-LSSVM	-	$74,6 \pm 2,8$		147,0	20,0%	
		$75,4 \pm 3,6$		92,0	50,0%	
		$74,3 \pm 3,7$		37,0	80,0%	
LSSVM	-	$75,1 \pm 3,9$	214	214,0	–	70%
IP-LSSVM		$72,2 \pm 0,0$		201,0	6,0%	
MOGAS-LSSVM	0,7	$76,3 \pm 4,7$		41,0	80,6%	
FSGAS-LSSVM	-	$74,7 \pm 3,7$		171,0	20,0%	
		$75,4 \pm 3,9$		107,0	50,0%	
		$75,8 \pm 4,5$		43,0	80,0%	
LSSVM	-	$73,8 \pm 5,0$	245	245,0	–	80%
IP-LSSVM		$70,6 \pm 4,9$		223,0	9,0%	
MOGAS-LSSVM	0,2	$76,9 \pm 4,5$		75,0	69,4%	
FSGAS-LSSVM	-	$74,3 \pm 4,5$		196,0	20,0%	
		$75,4 \pm 6,2$		122,0	50,0%	
		$74,4 \pm 4,8$		49,0	80,0%	

3. Comparações para o problema BCW

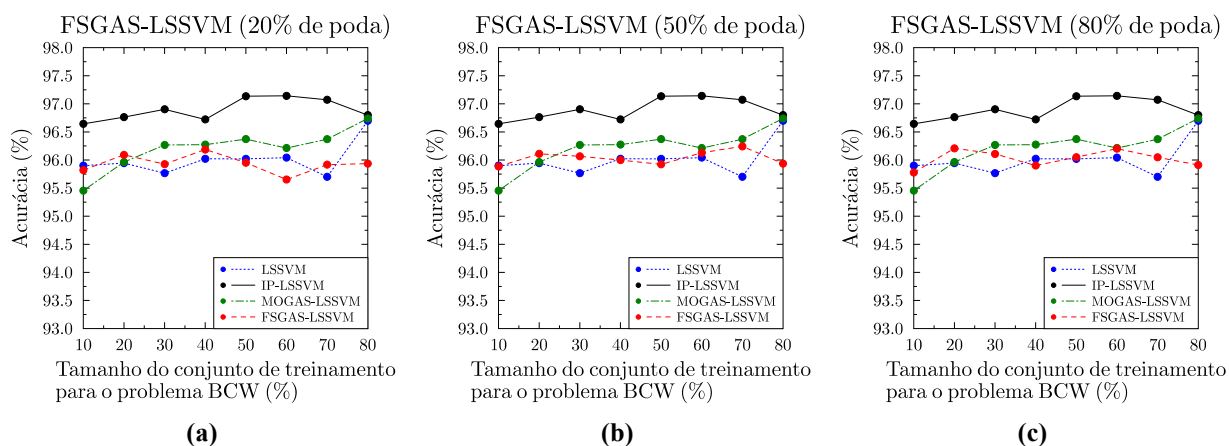


Figura 30: Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema BCW. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.

Diferentemente dos resultados observados nas demais simulações, para o problema BCW, os classificadores avaliados LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM apresentam desempenhos equiparáveis em termos de acurácia.

Na Figura 30a, considerando 60% do conjunto de dados para treinamento, o maior valor de acurácia alcançado é observado para o classificador IP-LSSVM ($97,1\% \pm 0,0$). Já o menor valor, é observado para o classificador FSGAS-LSSVM ($95,6 \pm 1,0$) com um percentual de poda definido para 20%. Levando em consideração os valores de desvio padrão, ratifica-se o equilíbrio em termos de acurácia entre os classificadores.

A conclusão a partir da análise feita relacionada à Figura 30a, também pode se estender aos resultados apresentados nas Figuras 30b e 30c. Mesmo utilizando diferentes tamanhos de conjuntos de dados para treinamento, os classificadores avaliados apresentam valores de acurácia muito próximos.

Também é possível notar uma certa estabilidade quanto a acurácia obtida ao longo da variação do tamanho do conjunto de dados para treinamento. Além da característica citada anteriormente e particular ao classificador FSGAS-LSSVM, mesmo variando o percentual de poda do classificador, seu desempenho também não é afetado. A exemplo, para 80% dos dados utilizados no processo de treinamento, foram obtidos $95,9\% \pm 1,6$, $95,9\% \pm 1,8$ e $95,9\% \pm 1,6$ respectivamente para 20%, 50% e 80% de poda.

A Tabela 14 apresenta as métricas de desempenho do conjunto de teste.

Tabela 14: Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% até 80% do conjunto de dados para treinamento considerando o problema BCW ($\gamma = 0,04$).

Modelo	θ	Taxa de Acerto	# PT	# VS's	Red	# DT.
LSSVM	-	$95,9 \pm 0,6$	68	68,0	—	10%
IP-LSSVM		$96,6 \pm 0,0$		35,0	48,5%	
MOGAS-LSSVM	0,0	$95,4 \pm 0,9$		34,0	50,4%	
FSGAS-LSSVM	-	$95,8 \pm 0,9$		54,0	20,0%	
		$95,8 \pm 0,6$		34,0	50,0%	
		$95,7 \pm 0,6$		14,0	80,0%	
LSSVM	-	$95,9 \pm 0,4$	137	137,0	—	20%
IP-LSSVM		$96,7 \pm 0,0$		70,0	48,9%	
MOGAS-LSSVM	0,0	$95,9 \pm 0,5$		68,0	50,3%	
FSGAS-LSSVM	-	$96,0 \pm 0,6$		110,0	20,0%	
		$96,1 \pm 0,5$		68,0	50,0%	
		$96,2 \pm 0,6$		27,0	80,0%	
LSSVM	-	$95,7 \pm 0,5$	205	205,0	—	30%
IP-LSSVM		$96,9 \pm 0,0$		105,0	48,7%	
MOGAS-LSSVM	0,3	$96,2 \pm 0,5$		24,0	88,3%	
FSGAS-LSSVM	-	$95,9 \pm 0,5$		164,0	20,0%	
		$96,0 \pm 0,6$		102,0	50,0%	
		$96,1 \pm 0,5$		41,0	80,0%	
LSSVM	-	$96,0 \pm 0,6$	273	273,0	—	40%
IP-LSSVM		$96,9 \pm 0,0$		140,0	48,7%	
MOGAS-LSSVM	0,2	$96,2 \pm 0,6$		49,0	82,0%	
FSGAS-LSSVM	-	$96,1 \pm 0,6$		218,0	20,0%	
		$96,0 \pm 0,7$		136,0	50,0%	
		$95,9 \pm 0,5$		55,0	80,0%	
LSSVM	-	$96,0 \pm 0,8$	342	342,0	—	50%
IP-LSSVM		$97,1 \pm 0,0$		175,0	48,8%	
MOGAS-LSSVM	0,0	$96,3 \pm 0,7$		167,0	51,1%	
FSGAS-LSSVM	-	$95,9 \pm 0,8$		274,0	20,0%	
		$95,9 \pm 0,8$		171,0	50,0%	
		$96,0 \pm 0,7$		68,0	80,0%	
LSSVM	-	$96,0 \pm 0,9$	410	410,0	—	60%
IP-LSSVM		$97,1 \pm 0,0$		210,0	48,7%	
MOGAS-LSSVM	0,0	$96,2 \pm 1,0$		202,0	50,7%	
FSGAS-LSSVM	-	$95,6 \pm 1,0$		328,0	20,0%	
		$96,1 \pm 0,9$		205,0	50,0%	
		$96,2 \pm 0,9$		82,0	80,0%	
LSSVM	-	$95,7 \pm 1,1$	478	478,0	—	70%
IP-LSSVM		$97,0 \pm 0,0$		245,0	48,7%	
MOGAS-LSSVM	0,3	$96,3 \pm 1,1$		120,0	74,8%	
FSGAS-LSSVM	-	$95,9 \pm 1,4$		382,0	20,0%	
		$96,2 \pm 1,1$		239,0	50,0%	
		$96,0 \pm 1,0$		96,0	80,0%	
LSSVM	-	$96,7 \pm 1,1$	546	546,0	—	80%
IP-LSSVM		$96,8 \pm 1,1$		284,0	48,0%	
MOGAS-LSSVM	0,3	$96,7 \pm 1,4$		162,0	70,1%	
FSGAS-LSSVM	-	$95,9 \pm 1,6$		437,0	20,0%	
		$95,9 \pm 1,8$		273,0	50,0%	
		$95,9 \pm 1,6$		109,0	80,0%	

4. Comparações para o problema PID

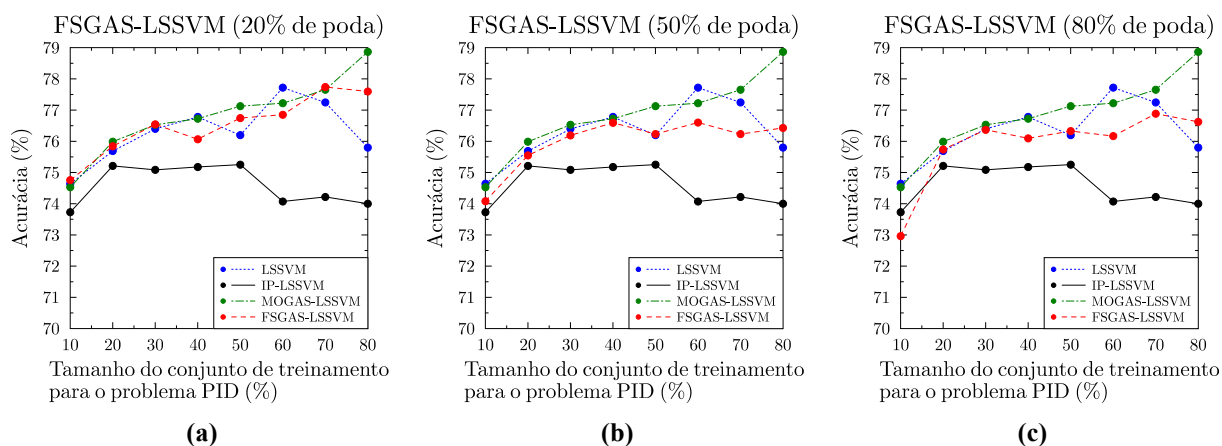


Figura 31: Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema PID. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.

Na Figura 31a entre 10% e 30% dos dados para treinamento, os classificadores LSSVM, MOGAS-LSSVM e FSGAS-LSSVM alcançam valores de acurácia compatíveis. Para o caso em que 40% dos dados são utilizados para treinamento, o classificador FSGAS-LSSVM apresenta uma leve perda de desempenho. Considerando 80% dos dados para treinamento, o classificador MOGAS-LSSVM apresenta-se superior, seguido do classificador FSGAS-LSSVM e LSSVM padrão. Para todos os casos, o classificador IP-LSSVM apresenta-se inferior aos demais.

Nas condições apresentadas nas Figuras 31b e 31c, de um modo em geral, o classificador FSGAS-LSSVM apresenta um funcionamento estável. A maior diferença que pode ser observada, está presente quando 10% do conjunto de dados é utilizado no processo de treinamento. Com 50% de poda, o classificador em questão alcança $74,0\% \pm 2,7$, valor superior a acurácia de $72,9\% \pm 2,3$ obtida com 80% de poda.

Ainda relacionado as Figuras 31b e 31c, para valores acima de 20% do conjunto de dados para treinamento, o classificador MOGAS-LSSVM apresenta-se levemente superior aos demais classificadores, com destaque maior em 80%, obtendo $78\% \pm 2,7$. Considerando 60% dos dados para treinamento, o classificador LSSVM padrão obtém $77,7\% \pm 1,9$ de acurácia enquanto o classificador MOGAS-LSSVM obtém $77,2\% \pm 1,8$.

A Tabela 15 apresenta as métricas de desempenho do conjunto de teste.

Tabela 15: Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% até 80% do conjunto de dados para treinamento considerando o problema PID ($\mu = 0,04$).

Modelo	θ	Taxa de Acerto	# PT	# VS's	Red	# DT.
LSSVM	-	$74,6 \pm 2,1$	77	77,0	—	10%
IP-LSSVM		$73,7 \pm 0,0$		73,0	5,1%	
MOGAS-LSSVM	0,0	$74,5 \pm 1,9$		33,0	57,2%	
FSGAS-LSSVM	-	$74,7 \pm 1,8$		62,0	20,0%	
		$74,0 \pm 2,7$		38,0	50,0%	
		$72,9 \pm 2,3$		15,0	80,0%	
LSSVM	-	$75,6 \pm 1,3$	154	154,0	—	20%
IP-LSSVM		$75,2 \pm 0,0$		146,0	7,1%	
MOGAS-LSSVM	0,0	$75,9 \pm 1,4$		68,0	55,8%	
FSGAS-LSSVM	-	$75,8 \pm 1,3$		123,0	20,0%	
		$75,5 \pm 1,7$		77,0	50,0%	
		$75,7 \pm 1,0$		31,0	80,0%	
LSSVM	-	$76,3 \pm 1,2$	230	230,0	—	30%
IP-LSSVM		$75,0 \pm 0,0$		218,0	5,2%	
MOGAS-LSSVM	0,5	$76,5 \pm 1,3$		49,0	78,7%	
FSGAS-LSSVM	-	$76,5 \pm 1,1$		184,0	20,0%	
		$76,1 \pm 1,1$		115,0	50,0%	
		$76,3 \pm 1,4$		46,0	80,0%	
LSSVM	-	$76,7 \pm 1,5$	307	307,0	—	40%
IP-LSSVM		$75,1 \pm 0,0$		292,0	4,8%	
MOGAS-LSSVM	0,2	$76,7 \pm 1,3$		96,0	68,8%	
FSGAS-LSSVM	-	$76,0 \pm 1,5$		246,0	20,0%	
		$76,5 \pm 1,3$		153,0	50,0%	
		$76,0 \pm 1,7$		61,0	80,0%	
LSSVM	-	$76,1 \pm 1,5$	384	384,0	—	50%
IP-LSSVM		$75,2 \pm 0,0$		365,0	4,9%	
MOGAS-LSSVM	0,2	$77,1 \pm 1,7$		131,0	65,9%	
FSGAS-LSSVM	-	$76,7 \pm 1,9$		307,0	20,0%	
		$76,2 \pm 1,8$		192,0	50,0%	
		$76,3 \pm 1,6$		77,0	80,0%	
LSSVM	-	$77,7 \pm 1,9$	461	461,0	—	60%
IP-LSSVM		$74,0 \pm 0,0$		438,0	4,9%	
MOGAS-LSSVM	1,0	$77,2 \pm 1,8$		125,0	72,8%	
FSGAS-LSSVM	-	$76,8 \pm 1,9$		369,0	20,0%	
		$76,6 \pm 2,3$		230,0	50,0%	
		$76,1 \pm 2,8$		92,0	80,0%	
LSSVM	-	$77,2 \pm 1,6$	538	538,0	—	70%
IP-LSSVM		$74,2 \pm 0,0$		511,0	5,0%	
MOGAS-LSSVM	0,8	$77,6 \pm 2,5$		163,0	69,6%	
FSGAS-LSSVM	-	$77,7 \pm 2,5$		430,0	20,0%	
		$76,2 \pm 2,5$		269,0	50,0%	
		$76,8 \pm 2,6$		108,0	80,0%	
LSSVM	-	$75,8 \pm 2,7$	614	614,0	—	80%
IP-LSSVM		$74,0 \pm 5,3$		503,0	18,1%	
MOGAS-LSSVM	0,3	$78,8 \pm 2,7$		238,0	61,2%	
FSGAS-LSSVM	-	$77,5 \pm 3,7$		491,0	20,0%	
		$76,4 \pm 3,3$		307,0	50,0%	
		$76,6 \pm 3,1$		123,0	80,0%	

5. Comparações para o problema RIP

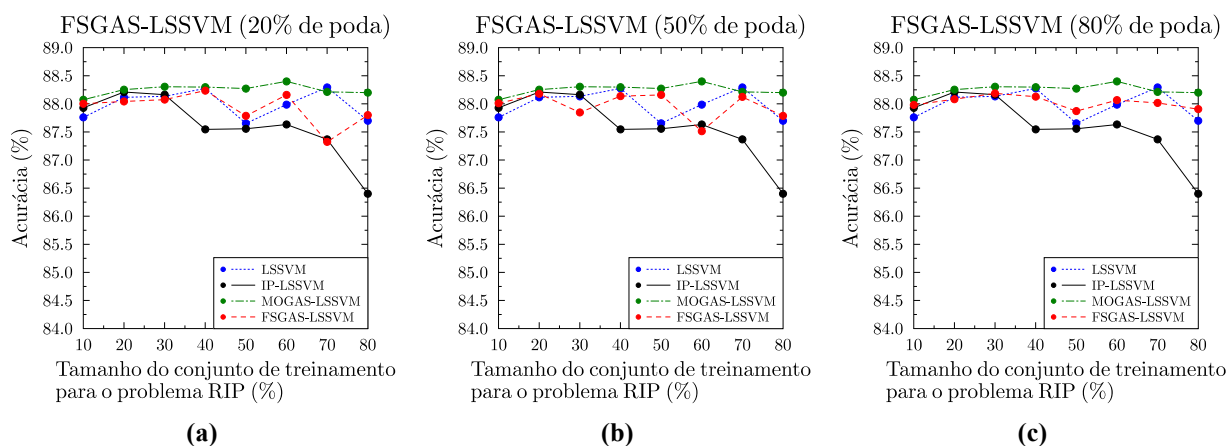


Figura 32: Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema RIP. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.

Na Figura 32a até 30% dos dados para treinamento, todos os classificadores apresentam valores de acurácia equiparáveis. Para 40% dos dados é possível notar que o desempenho do modelo IP-LSSVM diminui, em seguida, mantém-se estável até 60% e novamente decai, mostrando-se inferior aos demais classificadores considerando um conjunto de dados para treinamento de 80%.

Já o classificador FSGAS-LSSVM quando comparado ao modelo MOGAS-LSSVM, mostra-se inferior ao utilizar 50%, 70% e 80% dos dados para treinamento, tanto em termos de desempenho como considerando a redução de vetores-suporte obtida (dados disponíveis na Tabela 16).

Pode-se observar na Figura 32b que até 40% do conjunto de dados utilizado para treinamento, os classificadores apresentam um desempenho similar ao ilustrado na Figura 32a. Pode-se inferir que a mudança mais significativa consiste no desempenho do classificador FSGAS-LSSVM, que se mostra inferior aos demais modelos equiparando-se ao classificador IP-LSSVM.

Nas condições apresentadas na Figura 32c, de um modo em geral, pode-se notar que os classificadores LSSVM padrão, MOGAS-LSSVM e FSGAS-LSSVM são equiparáveis em termos de acurácia e que a partir de 40% dos dados para treinamento o classificador IP-LSSVM novamente apresenta-se inferior aos demais.

A Tabela 16 apresenta as métricas de desempenho do conjunto de teste.

Tabela 16: Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% até 80% do conjunto de dados para treinamento considerando o problema RIP ($\nu = 0,04$).

Modelo	θ	Taxa de Acerto	# PT	# VS's	Red	# DT.
LSSVM	-	$87,7 \pm 0,8$	125	125,0	–	10%
IP-LSSVM		$87,9 \pm 0,0$		119,0	4,8%	
MOGAS-LSSVM	0,0	$88,0 \pm 0,5$		125,0	58,5%	
FSGAS-LSSVM	-	$88,0 \pm 0,6$		100,0	20,0%	
		$88,0 \pm 0,5$		62,0	50,0%	
		$87,9 \pm 0,5$		25,0	80,0%	
LSSVM	-	$88,1 \pm 0,5$	250	250,0	–	20%
IP-LSSVM		$88,2 \pm 0,0$		237,0	5,2%	
MOGAS-LSSVM	0,9	$88,2 \pm 0,5$		42,0	83,32%	
FSGAS-LSSVM	-	$88,0 \pm 0,6$		200,0	20,0%	
		$88,1 \pm 0,6$		125,0	50,0%	
		$88,0 \pm 0,6$		50,0	80,0%	
LSSVM	-	$88,1 \pm 0,5$	375	375,0	–	30%
IP-LSSVM		$88,1 \pm 0,0$		356,0	5,0%	
MOGAS-LSSVM	1,0	$88,3 \pm 0,6$		83,0	77,5%	
FSGAS-LSSVM	-	$88,0 \pm 0,7$		300,0	20,0%	
		$87,8 \pm 0,8$		187,0	50,0%	
		$88,1 \pm 0,6$		75,0	80,0%	
LSSVM	-	$88,2 \pm 0,8$	500	500,0	–	40%
IP-LSSVM		$87,5 \pm 0,0$		475,0	5,0%	
MOGAS-LSSVM	0,1	$88,2 \pm 0,7$		181,0	63,7%	
FSGAS-LSSVM	-	$88,2 \pm 0,7$		400,0	20,0%	
		$88,1 \pm 0,6$		250,0	50,0%	
		$88,1 \pm 0,7$		100,0	80,0%	
LSSVM	-	$87,6 \pm 1,0$	625	625,0	–	50%
IP-LSSVM		$87,5 \pm 0,0$		594,0	4,9%	
MOGAS-LSSVM	0,4	$88,2 \pm 0,9$		207,0	66,8%	
FSGAS-LSSVM	-	$87,7 \pm 0,9$		500,0	20,0%	
		$88,1 \pm 1,1$		312,0	50,0%	
		$87,8 \pm 0,8$		125,0	80,0%	
LSSVM	-	$87,9 \pm 1,2$	750	750,0	–	60%
IP-LSSVM		$87,6 \pm 0,0$		712,0	5,0%	
MOGAS-LSSVM	0,7	$88,4 \pm 1,4$		238,0	68,2%	
FSGAS-LSSVM	-	$88,1 \pm 0,9$		600,0	20,0%	
		$87,5 \pm 1,1$		375,0	50,0%	
		$88,0 \pm 0,8$		150,0	80,0%	
LSSVM	-	$88,2 \pm 1,1$	875	875,0	–	70%
IP-LSSVM		$87,3 \pm 0,0$		831,0	5,0%	
MOGAS-LSSVM	0,3	$88,2 \pm 1,2$		330,0	62,2%	
FSGAS-LSSVM	-	$87,3 \pm 1,5$		700,0	20,0%	
		$88,1 \pm 1,2$		437,0	50,0%	
		$88,0 \pm 1,5$		175,0	80,0%	
LSSVM	-	$87,7 \pm 1,6$	1000	1000,0	–	80%
IP-LSSVM		$86,4 \pm 3,3$		496,0	56,4%	
MOGAS-LSSVM	0,9	$88,2 \pm 0,0$		350,0	65,0%	
FSGAS-LSSVM	-	$87,8 \pm 2,2$		800,0	20,0%	
		$87,7 \pm 1,9$		500,0	50,0%	
		$87,9 \pm 2,0$		200,0	80,0%	

6. Comparações para o problema BAN

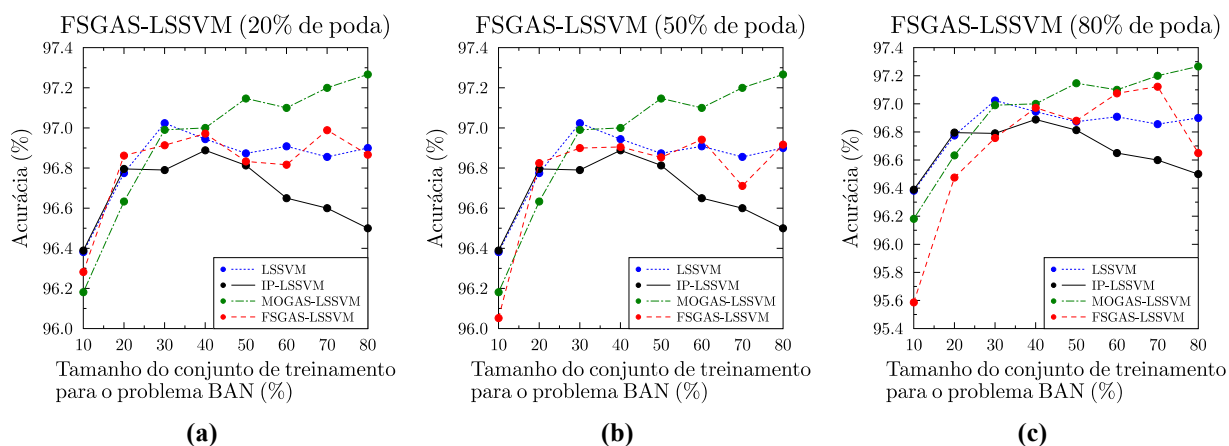


Figura 33: Comparação entre acurácia obtida pelos classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM em função do tamanho do conjunto de treinamento para o problema BAN. (a) classificador FSGAS-LSSVM com percentual de poda definido para 20%. (b) classificador FSGAS-LSSVM com percentual de poda definido para 50%. (c) classificador FSGAS-LSSVM com percentual de poda definido para 80%.

De um modo em geral, a Figura 33 mostra que os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM, quando submetidos ao problema BAN, possuem desempenhos equiparáveis. Os valores de acurácia obtidos não apresentam diferenças significativas.

Das Figuras 33a, 33b e 33c pode-se afirmar que o classificador MOGAS-LSSVM se destaca em termos de acurácia para valores acima de 40% dos dados para treinamento e o classificador IP-LSSVM apresenta-se inferior aos demais modelos para valores acima de 30% dos dados para treinamento.

O classificador LSSVM padrão é notadamente equiparável ao modelo MOGAS-LSSVM até cerca de 40% dos dados para treinamento. Em seguida, o desempenho observado permanece estável, sendo ainda superior ao obtido pelo classificador IP-LSSVM.

Na Figura 33c nota-se que o classificador FSGAS-LSSVM apresenta um desempenho inferior aos demais modelos até cerca de 30% dos dados utilizados para treinamento. Outra característica particular a essa situação é observada quando se considera um conjunto de treinamento acima de 70%, onde o desempenho obtido pelo modelo FSGAS-LSSVM equipara-se ao obtido pelo classificador IP-LSSVM.

A Tabela 17 apresenta as métricas de desempenho do conjunto de teste.

Tabela 17: Comparação entre os classificadores LSSVM, IP-LSSVM, MOGAS-LSSVM e FSGAS-LSSVM com 10% até 80% do conjunto de dados para treinamento considerando o problema BAN ($\nu = 0,04$).

Modelo	θ	Taxa de Acerto	# PT	# VS's	Red	# DT.
LSSVM	-	$96,3 \pm 0,5$	100	100,0	—	10%
IP-LSSVM		$96,3 \pm 0,0$		50,0	50,0%	
MOGAS-LSSVM	0,0	$96,1 \pm 0,5$		48,0	51,7%	
FSGAS-LSSVM	-	$96,2 \pm 0,6$		80,0	20,0%	
		$96,0 \pm 0,7$		50,0	50,0%	
		$95,5 \pm 0,7$		20,0	80,0%	
LSSVM	-	$96,7 \pm 0,4$	200	200,0	—	20%
IP-LSSVM		$96,7 \pm 0,0$		100,0	50,0%	
MOGAS-LSSVM	0,0	$96,6 \pm 0,3$		98,0	50,9%	
FSGAS-LSSVM	-	$96,8 \pm 0,3$		160,0	20,0%	
		$96,8 \pm 0,4$		100,0	50,0%	
		$96,4 \pm 0,4$		40,0	80,0%	
LSSVM	-	$97,0 \pm 0,4$	300	300,0	—	30%
IP-LSSVM		$96,7 \pm 0,0$		150,0	50,0%	
MOGAS-LSSVM	0,2	$96,9 \pm 0,4$		58,0	80,7%	
FSGAS-LSSVM	-	$96,9 \pm 0,4$		240,0	20,0%	
		$96,8 \pm 0,3$		150,0	50,0%	
		$96,7 \pm 0,3$		60,0	80,0%	
LSSVM	-	$96,9 \pm 0,6$	400	400,0	—	40%
IP-LSSVM		$96,8 \pm 0,0$		200,0	50,0%	
MOGAS-LSSVM	0,9	$97,0 \pm 0,3$		81,0	79,8%	
FSGAS-LSSVM	-	$96,9 \pm 0,4$		320,0	20,0%	
		$96,9 \pm 0,4$		200,0	50,0%	
		$96,9 \pm 0,5$		320,0	80,0%	
LSSVM	-	$96,8 \pm 0,5$	501	501,0	—	50%
IP-LSSVM		$96,8 \pm 0,0$		250,0	50,0%	
MOGAS-LSSVM	0,5	$97,1 \pm 0,5$		119,0	76,2%	
FSGAS-LSSVM	-	$96,8 \pm 0,5$		401,0	20,0%	
		$96,8 \pm 0,5$		250,0	50,0%	
		$96,8 \pm 0,6$		100,0	80,0%	
LSSVM	-	$96,9 \pm 0,7$	601	601,0	—	60%
IP-LSSVM		$96,6 \pm 0,0$		300,0	50,0%	
MOGAS-LSSVM	1,0	$97,1 \pm 0,5$		152,0	74,8%	
FSGAS-LSSVM	-	$96,8 \pm 0,5$		481,0	20,0%	
		$96,9 \pm 0,5$		300,0	50,0%	
		$97,0 \pm 0,5$		120,0	80,0%	
LSSVM	-	$96,8 \pm 0,8$	701	701,0	—	70%
IP-LSSVM		$96,6 \pm 0,0$		350,0	50,0%	
MOGAS-LSSVM	0,9	$97,2 \pm 0,8$		191,0	72,7%	
FSGAS-LSSVM	-	$96,9 \pm 0,8$		561,0	20,0%	
		$96,7 \pm 0,7$		350,0	50,0%	
		$97,1 \pm 0,8$		140,0	80,0%	
LSSVM	-	$96,9 \pm 1,3$	801	801,0	—	80%
IP-LSSVM		$96,5 \pm 1,4$		394,0	50,8%	
MOGAS-LSSVM	0,8	$97,2 \pm 1,2$		244,0	69,5%	
FSGAS-LSSVM	-	$96,8 \pm 1,0$		641,0	20,0%	
		$96,9 \pm 1,0$		400,0	50,0%	
		$96,6 \pm 0,9$		160,0	80,0%	

5.3 Conclusão

Neste capítulo apresenta-se a metodologia utilizada para avaliação dos classificadores propostos bem como as características das bases de dados: *Vertebral Column Pathologies*, *Haberman*, *Breast Cancer Wisconsin*, *Pima Indians Diabetes*, *Ripley* e *Banana*. Apresenta-se ainda as métricas de desempenho utilizadas para comparar as propostas apresentadas nesta dissertação com classificadores clássicos quando submetidos as bases de dados utilizadas nas simulações. Além da acurácia sobre o conjunto de teste e a redução média de vetores-suporte, foram apresentadas tabelas comparando o desempenho dos classificadores em diferentes situações como variação do tamanho do conjunto de treinamento e diferentes custos de poda. O principal objetivo da realização de testes em diversas condições é comprovar a robustez dos classificadores propostos e embasar as conclusões apresentadas no Capítulo 6.

CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo tem por objetivo apresentar as conclusões e considerações finais acerca dos métodos propostos obtidos durante o desenvolvimento desta dissertação. Além disso, são apresentadas as considerações finais sobre o atendimento aos objetivos descritos na introdução e a apresentação de propostas para continuidade deste trabalho.

6.1 Conclusões

Através da utilização de restrições de igualdade ao invés de desigualdade e pela incorporação da soma das variáveis de folga ao quadrado ponderadas por um termo de regularização, permitiu-se que a partir do classificador SVM, um novo classificador seja obtido a partir da resolução de um sistema de equações lineares ao invés da resolução de problemas de programação quadrática. Em contrapartida, uma das maiores vantagens do classificador SVM foi perdida: a esparsidade de multiplicadores de Lagrange. Diante disso, várias abordagens ao longo dos anos foram propostas com o intuito de se impor esparsidade ao modelo LSSVM.

Técnicas baseadas em Algoritmos Genéticos têm sido largamente aplicadas a otimização dos hiperparâmetros de classificadores de um modo em geral. Entretanto, a utilização dessas meta-heurísticas com foco na redução do número de vetores-suporte de classificadores LSSVM, têm sido pouco difundida na literatura atual. Apenas um trabalho com tal propósito foi encontrado, veja Seção 4.3.1.

Neste trabalho são propostos dois novos classificadores esparsos com base em Algoritmos Genéticos para o modelo LSSVM. A primeira proposta chamada de *Multi-Objective Genetic Algorithm for Sparse LSSVM*, *MOGAS-LSSVM*, consiste em uma nova abordagem para a construção de modelos LSSVM com base em AGs multiobjetivos com escalarização para tarefas de classificação.

A otimização realizada pelo classificador *MOGAS-LSSVM* a fim de decidir se um determinado padrão é ou não utilizado como vetor-suporte é guiada tanto pela acurácia

de treinamento (ou validação) como pela redução de vetores-suporte. Para esta proposta multiobjetivo, propõe-se também uma maneira de equilibrar os valores de acurácia e redução através de uma função de aptidão com base em escalarização (DUBOIS-LACOSTE; LÓPEZ-IBÁÑEZ; STÜTZLE, 2013), nos quais problemas multiobjetivos são transformados em um conjunto de problemas com apenas um único objetivo.

A segunda proposta, também baseada em Algoritmos Genéticos, intitulada *Fixed-Size Genetic Algorithm for Sparse LSSVM*, *FSGAS-LSSVM*, realiza a remoção de vetores-suporte de acordo com um parâmetro previamente definido. Neste classificador, o AG trabalha para encontrar as melhores posições para remoção de vetores-suporte menos relevantes segundo um percentual de remoção previamente definido. Devido ao esquema de operadores genéticos utilizados, a quantidade de vetores-suporte permanece inalterada, por isso a definição do classificador como sendo de *tamanho fixo*.

Vale ressaltar que devido as particularidades necessárias para manter fixa a quantidade de vetores-suporte em classificadores FSGAS-LSSVM, uma nova abordagem de tradução de indivíduos com codificação binária para inteira e vice-versa é proposta como operador de reprodução. A operação de reprodução em si é realizada na codificação inteiro em uma etapa intermediária.

Com base nos resultados encontrados, conclui-se que ambos os classificadores MOGAS-LSSVM (multiobjetivo) e FSGAS-LSSVM (tamanho fixo) são capazes de obter conjuntos reduzidos de vetores-suporte quando comparados ao classificador LSSVM padrão e ainda manter ou mesmo aumentar o desempenho dos classificadores. Também conclui-se que a utilização de AGs com o objetivo de definir um conjunto reduzido de vetores-suporte sem afetar o desempenho do classificador seja ele multiobjetivo ou não, consiste em uma técnica bastante interessante em plataformas com recursos computacionais limitados.

Em termos de redução de vetores-suporte o classificador MOGAS-LSSVM supera os classificadores clássicos utilizados nas comparações. Verifica-se uma redução média superior a 78% para o problema VCP (*Abordagem I*) e cerca de 87% para o problema HAB (*Abordagem II*), enquanto os valores obtidos pelos demais classificadores são próximos a 50%. Verifica-se ainda que o desempenho médio obtido pelo classificador MOGAS-LSSVM equipara-se aos resultados obtidos pelos classificadores LSSVM padrão, GAS-LSSVM, IP-LSSVM e P-LSSVM, sendo em certos casos superior. A exemplo cita-se o problema RIP com 88,1% (*Abordagem I*) e HAB com 75,1% (*Abordagem II*).

Nota-se que o desempenho médio obtido pelo classificador FSGAS-LSSVM independente do percentual de redução de vetores-suporte definido, mantém-se superior ao desempenho

outrora obtido pelos demais classificadores, exceto para o problema BCW. Particularmente para o problema BCW os classificadores FSGAS-LSSVM, IP-LSSVM e P-LSSVM apresentam acurácias equiparáveis até cerca de 83% de redução de vetores-suporte, acima desse valor verifica-se uma queda de desempenho para os classificadores IP-LSSVM e P-LSSVM. Já o classificador proposto FSGAS-LSSVM mantém seu desempenho mesmo para percentuais de poda acima de 83%.

6.2 Contribuições deste Trabalho

As contribuições apresentadas neste trabalho estão relacionadas aos temas: Máquinas de Vetores-Suporte por Mínimos Quadrados, métodos de poda em classificadores LSSVM e Algoritmos Genéticos.

A revisão bibliográfica apresentada no Capítulo 2 permite ao leitor uma compreensão sobre as principais diferenças, vantagens e desvantagens particulares aos classificadores SVM e LSSVM.

No Capítulo 4 são apresentados os classificadores MOGAS-LSSVM e FSGAS-LSSVM. Tais modelos empregam o uso de Algoritmos Genéticos em classificadores LSSVM com o objetivo de poda de vetores-suporte. Ambas as propostas visam manter o desempenho obtido quando comparadas aos classificadores clássicos. Paralelamente, impõem esparsidade à solução encontrada através da remoção de vetores-suporte que apresentam informações irrelevantes para a construção da superfície ótima de separação entre as classes de um referido problema. Além disso, no Capítulo 4, é proposto um novo esquema para reprodução de indivíduos com codificação binária dos genes em que se realiza um codificação intermediária para inteiro a fim de aplicar operadores de reprodução com base em permutação.

Os desempenhos obtidos em termos de acurácia assim como os valores de redução de vetores-suporte apresentados no Capítulo 5, atestam que as propostas apresentadas satisfazem os objetivos que foram definidos inicialmente.

6.3 Trabalhos Futuros

Como trabalhos futuros pode-se sugerir a utilização do algoritmo multiobjetivo NSGA-II em substituição a técnica *a priori* de otimização multiobjetivo que utiliza escalarização, como a realizada nos classificadores MOGAS-LSSVM. Neste tipo de método emprega-se o conceito de *dominância de Pareto* onde soluções factíveis do problema são comparadas. As soluções não

dominadas formam a *Frenteira de Pareto*, que são as soluções ótimas encontradas disponíveis a cada rodada.

Outro trabalho sugerido consiste na geração de uma população inicial do AG otimizada. Sabe-se que o tempo computacional de um Algoritmo Genético pode ser reduzido, dentre outros fatores, através da geração de uma população inicial de alta diversidade. Espera-se que ao aplicar uma técnica desse tipo em um AG multiobjetivo, o tempo de convergência possa ser reduzido.

Sugere-se ainda, a modificação dos algoritmos propostos tornando-os classificadores multiclass. Atualmente, trabalha-se apenas com problemas binários. Entretanto, torna-se interessante estender a abordagem desses modelos para que problemas com mais de duas classes sejam contemplados. Uma sugestão seria realizar uma combinação de classificadores.

Por fim, outro trabalho interessante a ser considerado é a utilização de kernels não-lineares nos classificadores propostos a fim de comprovar a robustez dos novos modelos na resolução de problemas no espaço de características.

Bibliografia

AGGARWAL, S.; GARG, R.; GOSWAMI, P. A review paper on different encoding schemes used in genetic algorithms. In: **International Journal of Advanced Research in Computer Science and Software Engineering**. [S.l.: s.n.], 2014. v. 4.

ALVAREZ, D.; HORNERO, R.; MARCOS, J.V.; DEL CAMPO, F. Feature selection from nocturnal oximetry using genetic algorithms to assist in obstructive sleep apnoea diagnosis. **Medical Engineering & Physics**, v. 34, n. 8, p. 1049–1057, 2012.

ARABAS, J.; MICHALEWICZ, Z.; MULAWKA, J. GAVaPS-a genetic algorithm with varying population size. In: **Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on**. Orlando, FL: IEEE Press, 1994. p. 73–78 vol.1.

AYDIN, I.; KARAKOSE, M.; AKIN, E. A multi-objective artificial immune algorithm for parameter optimization in support vector machine. **Applied Soft Computing**, Elsevier, v. 11, n. 1, p. 120–129, 2011.

BAKER, J.E. Reducing bias and inefficiency in the selection algorithm. In: **Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application**. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987. p. 14–21. ISBN 0-8058-0158-8. Disponível em: < <http://dl.acm.org/citation.cfm?id=42512.42515> > .

BARBOSA, A.M.; RIBEIRO, L.C.; ARANTES, J.M.O. Algoritmo genético multiobjetivo: Sistema adaptativo com elitismo. In: **Brazilian Conference on Dynamics, Control and their Applications (DINCON)**. Serra Negra, SP: [s.n.], 2010.

BARCELLOS, J.C.H. **Algoritmos Genéticos Adaptativos: Um estudo comparativo**. 131 p. Dissertação (Mestrado) — Universidade de São Paulo, 2000. Escola Politécnica, Universidade de São Paulo, São Paulo.

BONESSO, D. **Estimacão dos Parâmetros do Kernel em um Classificador SVM na Classificação de Imagens Hiperespectrais em uma Abordagem Multiclasse**. 108 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul (UFRGS), 2013. Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia, Programa de Pós-Graduação em Sensoriamento Remoto. Porto Alegre.

BOSER, B.E.; GUYON, I.; VAPNIK, V. A training algorithm for optimal margin classifiers. In: **Proceedings of the Fifth Annual Workshop on Computational Learning Theory**. New York, NY: ACM Press, 1992. p. 144–152.

BRAIN, Z.E.; ADDICOAT, M.A. Using Meta-Genetic Algorithms to tune parameters of Genetic Algorithms to find lowest energy Molecular Conformers. In: **ALIFE**. Odense, Denmark: [s.n.], 2010. p. 378–385.

Multiobjective Optimization, Interactive and Evolutionary Approaches, v. 5252 de **Lecture Notes in Computer Science**, (Lecture Notes in Computer Science, v. 5252). Germany: Springer, 2008.

BURBIDGE, Robert; BUXTON, Bernard. An introduction to support vector machines for data mining. In: **KEYNOTE PAPERS, YOUNG OR12, UNIVERSITY OF NOTTINGHAM, OPERATIONAL RESEARCH SOCIETY, OPERATIONAL RESEARCH SOCIETY**. Cambridge, UK: Press, 2001. p. 3–15.

BURGES, C.J.C. Simplified support vector decision rules. In: **Proceedings of the 13th International Conference on Machine Learning (ICML'96)**. Bari, Italy: Morgan Kaufmann, 1996. p. 71–77. ISSN 1089-3555.

BURGES, C.J.C.; SCHOLKOPF, B. Improving the accuracy and speed of support vector machines. In: **Advances in Neural Information Processing Systems 9**. [S.l.]: MIT Press, 1997. p. 375–381.

CAMARGO, S.S. **Um Modelo Neural de Aprimoramento Progressivo para Redução de Dimensionalidade**. 107 p. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2010. Instituto de Informática, Programa de Pós-graduação em Computação, Porto Alegre.

CARDOSO, F.A.C.M. **Algoritmos Genéticos para Interpolação Não-Linear de Imagem e Decodificação de Códigos Lineares**. 149 p. Dissertação (Mestrado) — Universidade Estadual de Campinas (UNICAMP), 1998. Faculdade de Engenharia Elétrica e de Computação, Departamento de Comunicações, Campinas.

CARVALHO, B.P.R. **Novas Estratégias para Detecção Automática de Vetores de Suporte em Least Squares Support Vector Machines**. 142 p. Dissertação (Mestrado) — Universidade Federal de Minas Gerais (UFMG), 2005. Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica, Belo Horizonte.

CARVALHO, B.P.R.; BRAGA, A.P. IP-LSSVM: A two-step sparse classifier. **Pattern Recognition Letters**, v. 30, p. 1507–1515, 2009.

CARVALHO, B.P.R.; WILIAN, S.L.; ANTONIO, P.B. de. RRS + LS-SVM: a new strategy for “a priori” sample selection. **Neural Computing and Applications**, Springer-Verlag, v. 16, n. 3, p. 227–234, 2007.

CASTRO, R.E. **Otimização de estruturas com multi-objetivos via algoritmos genéticos**. 206 p. Tese (Doutorado) — Universidade Federal do Rio de Janeiro (UFRJ), 2001. Engenharia Civil, Programa de Pós-Graduação de Engenharia, Rio de Janeiro.

COELLO, C.A.C.; GREGORIO, T.P. A micro-genetic algorithm for multiobjective optimization. In: **Evolutionary Multi-Criterion Optimization**. London, UK: Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 1993). p. 126–140.

COELLO, C.A.C.; LAMONT, G.B.; VELDHUIZEN, D.A. Van. **Evolutionary Algorithms for Solving Multi-Objective Problems**. 2. ed. New York: Springer-Verlag, Inc., 2007.

CORTES, C.; VAPNIK, V. Support-vector networks. **Mach. Learn.**, Kluwer Academic Publishers, v. 20, p. 273–297, 1995.

DEB, K.; AGRAWAL, S.; PRATAP, A.; MEYARIVAN, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: **Parallel Problem Solving from Nature PPSN VI**. Paris, France: Springer Berlin Heidelberg, 2000, (Lecture Notes in Computer Science, v. 1917). p. 849–858.

DONG, S.; LUO, T. Bearing degradation process prediction based on the PCA and optimized LS-SVM model. **Measurement**, v. 46, n. 9, p. 3143 – 3152, 2013.

DOWNS, T.; GATES, K.E.; MASTERS, A. Exact simplification of support vector solutions. **Journal of Machine Learning Research**, v. 2, p. 293–297, 2001.

DUBOIS-LACOSTE, J.; LÓPEZ-IBÁÑEZ, M.; STÜTZLE, T. Combining Two Search Paradigms for Multi-objective Optimization: Two-Phase and Pareto Local Search. In: **Hybrid Metaheuristics**. [S.l.]: Springer Berlin Heidelberg, 2013, (Studies in Computational Intelligence, v. 434). p. 97–117.

EHRGOTT, M. **Multicriteria Optimization**. 2. ed. Berlin Heidelberg, NY: Lecture Notes in Economics and Mathematical Systems, 2005.

EIBEN, A.E.; SMITH, J.E. **Introduction to Evolutionary Computation**. [S.l.]: Springer, 2003.

FARAGÓ, P.; FARAGÓ, C.; HINTEA, S.; CÎRLUGEA, M. An Evolutionary Multi-objective Optimization Approach to Design the Sound Processor of a Hearing Aid. In: **International Conference on Advancements of Medicine and Health Care through Technology**. Cluj-Napoca, Romania: Springer International Publishing, 2014. v. 44, p. 181–186.

FERRAO, M.F.; MELLO, C.; BORIN, A.; MARETTO, D.A.; POPPI, R.J. LS-SVM: uma nova ferramenta quimiométrica para regressão multivariada. Comparação de modelos de regressão LS-SVM e PLS na quantificação de adulterantes em leite em pó empregando NIR. **Química Nova**, scielo, v. 30, p. 852–859, 2007. ISSN 0100-4042.

FOGEL, L.J. Autonomous automata. **Industrial Research**, v. 4, p. 14–19, 1962.

FONSECA, C.M.; FLEMING, P.J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: **Proceedings of the 5th International Conference on Genetic Algorithms**. [S.l.]: Morgan Kaufmann Publishers Inc., 1993. p. 416–423.

FREITAS, F.G.; MAIA, C.L.B.; CAMPOS, G.A.L.; SOUZA, J.T. Otimização em teste de software com aplicação de metaheurísticas. **Revista de Sistemas de Informação da FSMA**, n. 5, p. 3–13, 2010.

GARCIA, R.P. **Estratégias de penalização adaptativa para a solução de problemas de otimização com restrições via algoritmos genéticos**. 170 p. Dissertação (Mestrado) — Universidade Federal de Juiz de Fora (UFJF), 2014. Programa de Pós-graduação em Modelagem Computacional, Juiz de Fora.

GEEBELEN, D.; SUYKENS, J. A. K.; VANDEWALLE, J. Reducing the Number of Support Vectors of SVM Classifiers Using the Smoothed Separable Case Approximation. **IEEE Transactions on Neural Network and Learning Systems**, v. 23, n. 4, p. 682–688, 2012.

- GESTEL, T.V.; SUYKENS, J.A.K.; BAESSENS, B.; VIAENE, S.; VANTHIENEN, J.; DEDENE, G.; MOOR, B. de; VANDEWALLE, J. Benchmarking least squares support vector machine classifiers. **Machine Learning**, Kluwer Academic Publishers-Plenum Publishers, p. 5–32, 2004.
- GIJSBERTS, A.; METTA, G.; ROTHKRANTZ, L. Evolutionary optimization of least-squares support vector machines. In: **Data Mining**. [S.l.]: Springer US, 2010, (Annals of Information Systems, v. 8). p. 277–297.
- GOLDBERG, D.E. **Genetic algorithms in search, optimization & machine learning**. Boston, MA, USA: Addison-Wesley, 1989.
- GUNN, S.R. **Support Vector Machines for Classification and Regression**. [S.l.], 1998.
- HADAMARD, J. **Four Lectures on Mathematics**. New York: Columbia University Press, 1915.
- HADY, M.F.A.; HERBAWI, W.; WEBER, M.; SCHWENKER, F. A multi-objective genetic algorithm for pruning support vector machines. In: **Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on**. Boca Raton, FL: IEEE Computer Society, 2011. p. 269–275. ISSN 1082-3409.
- HAYKIN, S. **Redes Neurais: Princípios e práticas**. 2. ed. Porto Alegre: Bookman, 2001.
- HOEGAERTS, L.; SUYKENS, J.A.K.; VANDEWALLE, J.; MOOR, B. De. A comparison of pruning algorithms for sparse least squares support vector machines. In: **Neural Information Processing**. Calcutta, India: Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3316). p. 1247–1253.
- HOLLAND, J. **Adaptation in natural and artificial systems**. MA, USA: Ann Arbor, 1975.
- HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D.E. A niched pareto genetic algorithm for multiobjective optimization. In: **Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence. Proceedings of the First IEEE Conference on**. Orlando, FL: IEEE, 1994. p. 82–87 vol.1.
- HSU, C.W.; CHANG, C.C.; LIN, C.J. **A practical guide to support vector classification**. [S.l.], 2003.
- HU, T.; HARDING, S.; BANZHAF, W. Variable population size and evolution acceleration: A case study with a parallel evolutionary algorithm. **Genetic Programming and Evolvable Machines**, Kluwer Academic Publishers, Hingham, MA, USA, v. 11, n. 2, p. 205–225, jun 2010.
- HUANG, J.; HU, X.; YANG, F. Support vector machine with genetic algorithm for machinery fault diagnosis of high voltage circuit breaker. **Measurement**, v. 44, n. 6, p. 1018–1027, 2011.
- IYODA, E.M. **Inteligência Computacional no Projeto Automático de Redes Neurais Híbridas e Redes Neurofuzzy Heterogêneas**. 166 p. Dissertação (Mestrado) — Universidade Estadual de Campinas (UNICAMP), 2000. Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Computação e Automação Industrial, São Paulo.

JAUHARI, S.; RIZVI, S.A.M. Mining Gene Expression Data Focusing Cancer Therapeutics: A Digest. **Computational Biology and Bioinformatics, IEEE/ACM Transactions on**, v. 11, n. 3, p. 533–547, 2014.

KNOWLES, J.; CORNE, D. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: **Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on**. Washington, DC: IEEE, 1999. v. 1, p. –105 Vol. 1.

KOZA, J. R. Hierarchical genetic algorithms operating on populations of computer programs. In: **Proceedings of the Eleventh International Joint Conference on Artificial Intelligence IJCAI-89**. San Francisco, CA, USA: Morgan Kaufmann, 1989. v. 1, p. 768–774.

KUMAR, A. Encoding schemes in genetic algorithm. **International Journal of Advanced Research in IT and Engineering**, v. 2, n. 3, 2013.

LEE, Y.J.; MANGASARIAN, O.L. RSVM: Reduced support vector machines. In: **Data Mining Institute, Computer Sciences Department, University of Wisconsin**. Chigaco, IL, USA: [s.n.], 2001. p. 00–07.

LI, X.; KIRLEY, M. The effects of varying population density in a fine-grained parallel genetic algorithm. In: **Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on**. Honolulu, HI: IEEE Computer Society, 2002. v. 2, p. 1709–1714.

LI, Y.; LIN, C.; ZHANG, W. Improved sparse least-squares support vector machine classifiers. **Neurocomputing**, v. 69, n. 13-15, p. 1655–1658, 2006.

LICHMAN, M. **UCI Machine Learning Repository**. 2013. Disponível em: <<http://archive.ics.uci.edu/ml>> .

MALAQUIAS, N.G.L. **Uso dos Algoritmos Gen éticos para a Otimização de Rotas de Distribuição**. 97 p. Dissertação (Mestrado) — Universidade Federal de Uberlândia (UFU), 2006. Faculdade de Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica, Uberlândia.

MALL, R.; SUYKENS, J.A.K. Sparse reductions for fixed-size least squares support vector machines on large scale data. In: **Advances in Knowledge Discovery and Data Mining**. Gold Coast, Australia: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 7818). p. 161–173.

MARETTO, D.A. **Comparação entre máquinas de vetores de suporte por mínimos quadrados (LS-SVM) e métodos lineares para transferência de calibração**. 94 p. Dissertação (Mestrado) — Universidade Estadual de Campinas (UNICAMP), 2007. Instituto de Química, Campinas.

MERCER, J. Functions of positive and negative type, and their connection with the theory of integral equations. **Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences**, The Royal Society, v. 209, n. 441-458, p. 415–446, 1909. ISSN 0264-3952.

MINETTI, G.F.; ALFONSO, H.A. Variable size population in parallel evolutionary algorithms. In: **Intelligent Systems Design and Applications, 2005. ISDA '05. Proceedings. 5th International Conference on**. Wroclaw, Poland: IEEE, 2005. p. 350–355.

MITCHELL, M. **An Introduction to Genetic Algorithms**. MA, USA: MIT Press, 1998.

MOLINA, C.R.R. **Sparse methods for Online Support Vector Machine**. 50 p. Dissertação (Mestrado) — University of Barcelona, 2014. Master in Artificial Intelligence, Faculty of Mathematics, Barcelona.

MOUSAVI, S.M.; IRANMANESH, S.H. Least squares support vector machines with genetic algorithm for estimating costs in npd projects. In: **Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on**. Xi'an: IEEE, 2011. p. 127–131.

MUSTAFA, M.W.; SULAIMAN, M.H.; SHAREEF, H.; KHALID, S.N.A. Reactive power tracing in pool-based power system utilising the hybrid genetic algorithm and least squares SVM. **IET Generation, Transmission & Distribution**, v. 6, n. 2, p. 133–141, 2012.

NOSCHESE, S.; REICHEL, L. Inverse problems for regularization matrices. **Numerical Algorithms**, Springer US, Sardinia, Italy, v. 60, n. 4, p. 531–544, 2012. ISSN 1017-1398. Disponível em: <<http://dx.doi.org/10.1007/s11075-012-9576-8>>.

OSUNA, E.; FREUND, R.; GIROSI, F. An improved training algorithm for support vector machines. In: **Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop**. Amelia Island, FL: IEEE, 1997. p. 276–285. ISSN 1089-3555.

PADILHA, C.A.A. **Algoritmos Genéticos Aplicados a um Comitê de LS-SVM em Problemas de Classificação**. 61 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte (UFRN), 2013. Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal.

PALENCIA, A.E.R.; DELGADILLO, G.E.M. A computer application for a bus body assembly line using Genetic Algorithms. **International Journal of Production Economics**, v. 140, n. 1, p. 431–438, 2012.

PARREIRAS, R.O. **Algoritmos Evolucionários e Técnicas de Tomada de Decisão em Análise Multicritério**. 165 p. Tese (Doutorado) — Universidade Federal de Minas Gerais (UFMG), 2006. Programa de Pós-Graduação em Engenharia Elétrica, Escola de Engenharia da UFMG, Belo Horizonte.

PENROSE, R. A generalized inverse for matrices. **Mathematical Proceedings of the Cambridge Philosophical Society**, v. 51, p. 406–413, 7 1955. ISSN 1469-8064.

PERES, R.T.; PEDREIRA, C.E. Generalized risk zone: Selecting observations for classification. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 31, n. 7, p. 1331–1337, July 2009.

PLATT, J. **Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines**. [S.l.], 1998.

QASIM, A.R.; ELDOS, T. Population sizing scheme for genetic algorithm. In: **Computer Systems and Applications, 2007. AICCSA '07. IEEE/ACS International Conference on**. Amman: IEEE, 2007. p. 381–384.

REN, Y.; BAI, G. Determination of optimal svm parameters by using ga/pso. **Journal of Computers**, v. 5, n. 8, p. 1160–1168, 2010.

ROCHA NETO, A. R. **SINPATICO II: Novas Estratégias de Aprendizado de Máquina para Classificação de Patologias da Coluna Vertebral**. 158 p. Tese (Doutorado) — Universidade Federal do Ceará, 2011. Departamento de Engenharia de Teleinformática, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza.

ROCHA NETO, A. R.; BARRETO, G.A. Opposite maps: Vector quantization algorithms for building reduced-set svm and lssvm classifiers. **Neural Processing Letters**, v. 37, n. 1, p. 3–19, 2013.

RODRIGUES, C.F. **Análise Comparativa entre os Métodos Decomposição em Valores Singulares e Análise de Componentes, Principais Envolvendo Matrizes Esparsas de Grande Porte**. 171 p. Dissertação (Mestrado) — Universidade Federal de Minas Gerais (UFMG), 2011. Programa de Pós-Graduação em Ciência da Computação, Belo Horizonte.

RUDOLPH, G. Evolutionary search under partially ordered fitness sets. In: **In Proceedings of the International Symposium on Information Science Innovations In Engineering of Natural and Artificial Intelligent Systems (ISI 2001)**. [S.l.]: ICSC Academic Press, 2001. p. 818–822.

SAMADZADEGAN, F.; SOLEYMANI, A.; ABBASPOUR, R.A. Evaluation of genetic algorithms for tuning svm parameters in multi-class problems. In: **Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on**. Budapest: IEEE, 2010. p. 323–328.

SANTOS, G.S.; LUVIZOTTO, L.G.J.; MARIANI, V.C.; COELHO, L.S. Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process. **Expert Systems with Applications**, v. 39, n. 5, p. 4805–4812, 2012.

SCHAFFER, J. David. Multiple objective optimization with vector evaluated genetic algorithms. In: **Proceedings of the 1st International Conference on Genetic Algorithms**. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985. p. 93–100.

SCHWEFEL, H.P. **Evolutionsstrategie und numerische optimierung**. Tese (Doutorado) — Technical University of Berlin, 1975.

SILVA, D.A.; ROCHA NETO, A. R. Multi-objective genetic algorithms for sparse least square support vector machines. In: **Intelligent Data Engineering and Automated Learning - IDEAL 2014**. Salamanca, Spain: Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8669). p. 158–166.

SILVA, D.A.; ROCHA NETO, A. R. A genetic algorithms-based lssvm classifier for fixed-size set of support vectors. In: **International Work-Conference on Artificial Neural Networks - IWANN 2015**. Palma de Mallorca, Spain: Springer International Publishing, 2015, (Lecture Notes in Computer Science, v. 9095). p. 127–141.

SILVA, J.P.; ROCHA NETO, A. R. Sparse least squares support vector machines via genetic algorithms. In: **Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on**. Ipojuca: IEEE, 2013. p. 248–253.

- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evol. Comput.**, MIT Press, Cambridge, MA, USA, v. 2, n. 3, p. 221–248, 1994.
- SUYKENS, J.A.K.; GESTEL, T.V.; BRABANTER, J.; VANDEWALLE, J. **Least Squares Support Vector Machines**. Singapore: World Scientific, 2002.
- SUYKENS, J.A.K.; LUKAS, L.; VANDEWALLE, J. Sparse approximation using least squares support vector machines. In: **Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on**. Geneva: IEEE, 2000. v. 2, p. 757–760.
- SUYKENS, J.A.K.; LUKAS, L.; VANDEWALLE, J. Sparse least squares support vector machine classifiers. In: **European Symposium on Artificial Neural Networks - ESANN'2000**. Bruges, Belgium: [s.n.], 2000. p. 37–42.
- SUYKENS, J.A.K.; VANDEWALLE, J. Least squares support vector machine classifiers. **Neural Processing Letters**, Kluwer Academic Publishers, v. 9, p. 293–300, 1999.
- SUYKENS, J.A.K.; VANDEWALLE, J. Multiclass least squares support vector machines. In: **Neural Networks, 1999. IJCNN '99. International Joint Conference on**. Washington, DC: IEEE, 1999. v. 2, p. 900–903. ISSN 1098-7576.
- THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. 4th. ed. [S.l.]: Academic Press, 2008.
- VALYON, J.; HORVATH, G. A sparse least squares support vector machine classifier. In: **Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on**. [S.l.]: IEEE, 2004. v. 1, p. –548.
- VAPNIK, V. **Estimation of Dependences Based on Empirical Data**. New York, NY, USA: Springer-Verlag, Inc., 1982.
- VAPNIK, V. **The Nature of Statistical Learning Theory**. New York, NY, USA: Springer-Verlag, Inc., 1995.
- VAPNIK, V. **Statistical Learning Theory**. [S.l.]: John Wiley & Sons, 1998.
- VAPNIK, V.; LERNER, A. Pattern recognition using generalized portrait method. **Automation and Remote Control**, v. 24, 1963.
- VELDHUIZEN, D.A.V. **Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations**. Tese (Doutorado), Wright Patterson AFB, OH, USA, 1999.
- VELDHUIZEN, D.A.V.; LAMONT, G.B. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. **Evol. Comput.**, MIT Press, Cambridge, MA, USA, v. 8, n. 2, p. 125–147, jun 2000.
- YU, L.; CHEN, H.; WANG, S.; LAI, K.K. Evolving least squares SVM for stock market trend mining. **IEEE Trans. on Evolutionary Computation**, v. 13, n. 1, p. 87–102, 2009.
- ZHANG, X.; CHEN, X.; HE, Z. An aco-based algorithm for parameter optimization of support vector machines. **Expert Systems with Applications**, v. 37, n. 9, p. 6618–6628, 2010.
- ZITZLER, Eckart; LAUMANN, Marco; THIELE, Lothar. **SPEA2: Improving the Strength Pareto Evolutionary Algorithm**. Barcelona, Spain, 2001.