

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
DEPARTAMENTO DE TELEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

David Clifte da Silva Vieira

**Máquinas de Vetores-Suporte de Mínimos Quadrados Podadas Via
Métodos de Regressão Esparsos**

Fortaleza – CE

07 de Abril de 2017

David Clifte da Silva Vieira

Máquinas de Vetores-Suporte de Mínimos Quadrados Podadas Via Métodos de Regressão Esparsos

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Artificial

Orientador: Prof. Dr. Antonio Wendell de Oliveira Rodrigues

Coorientador: Prof. Dr. Ajalmar Rêgo da Rocha Neto

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

07 de Abril de 2017

DAVID CLIFTE DA SILVA VIEIRA

Máquinas de Vetores-Suporte de Mínimos Quadrados Podadas Via Métodos de
Regressão Esparsos

Dissertação submetida à Coordenação do Curso de Pós-graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação, área de concentração Ciência da Computação.

Aprovada em 07/04/2017.

BANCA EXAMINADORA

Prof. Dr. Antonio Wendell de Oliveira Rodrigues (Orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE

Prof. Dr. Ajalmar Rêgo da Rocha Neto (Co-orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE

Prof. Dr. Pedro Pedrosa Rebouças Filho
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE

Prof. Dr. Arthur Plínio de Souza Braga
Universidade da Federal do Ceará - UFC

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)

- V657m Vieira, David Clifte da Silva Vieira.
 Máquinas de Vetores-Suporte de Mínimos Quadrados Podadas Via Métodos de Regressão Esparsos /
David Clifte da Silva Vieira Vieira. - 2017.
 103 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus
Fortaleza, 2017.
 Orientação: Prof. Dr. Antonio Wendell de Oliveira Rodrigues.
 Coorientação: Prof. Dr. Ajalmar Rêgo da Rocha Neto.
1. Máquinas de Vetores Suporte. 2. Máquinas de Vetores Suporte de Mínimos Quadrados. 3. Solução
Esparsa. 4. Regressão Esparsa. I. Título.

CDD 004

Dedico a Deus, pela força e determinação durante este projeto.

Agradecimentos

A Deus, por todas as coisas boas e más que me aconteceram. Cada uma delas, ao seu modo, as quais me fizeram chegar onde eu cheguei, e me fizeram ser quem eu sou. Foi a minha jornada de tropeços, vitórias e derrotas que me fez enxergar o verdadeiro significado e beleza da vida.

Aos professores Dr. Wendell e Dr. Ajalmar pelo papel de orientadores, os seus ensinamentos foram muito além dos conteúdos do currículo. Vocês me deram uma oportunidade, acreditaram em mim e nas minhas capacidades, e por tudo isso lhes agradeço. Mas agradeço também por tudo que me ensinaram, pela paciência que demonstram no início e por nunca terem desistido de mim. O meu mais sincero obrigado.

Ao professor coordenador do programa, Dr. Pedrosa, pelo apoio e sugestões apresentadas, além de manter todos os alunos do programa motivados, ressaltando as qualidades e deveres que qualquer pesquisador deve ter.

A todos os professores do programa do PPGCC, os quais foram importantes no desenvolvimento das habilidades necessárias para o desenvolvimento deste trabalho. Muito obrigado pela dedicação, paciência e carinho ao lecionar.

Ao professor Dr. Geraldo Ramalho (IFCE), por ter me apresentado o ambiente de pesquisa e me orientado no período de bolsa de iniciação científica.

Aos meus pais, se há algo que faz diferença na formação da personalidade e na vida de uma pessoa é o amor que ela recebe. Vocês me educaram com amor, dedicaram-se à minha educação como ser humano, me deram amor. Vocês fizeram de mim a pessoa que hoje sou, e eu só tenho motivos para agradecer.

À Renata Passos, fiel companheira que sempre me apoiou em todos os momentos, principalmente nos mais difíceis. Quero lhe agradecer por suportar meus defeitos, tolerar meus humores e, principalmente, por me entender.

A todos os companheiros de curso, em especial, Madson, Lucas, Roger, aos companheiros do curso do programa de mestrado em telecomunicações do IFCE, em especial, Alan, Elizangela, Saulo e Shara, obrigado pelo apoio e as discussões frutíferas e infrutíferas, necessárias para o bom convívio e desenvolvimento de cada um.

"O primeiro gole do copo das ciências naturais pode até torná-lo ateu. Mas, no fundo do mesmo copo, Deus o aguarda.", Werner Heisenberg, Prêmio Nobel de Física (1901-1976).

Resumo

Máquinas de Vetores Suporte de Mínimos Quadrados (*Least Squares Support Vector Machines*, LSSVM) são uma alternativa para as Máquinas de Vetores Suporte (*Support Vector Machines*, SVM) convencionais, pois o seu processo de treinamento é baseado em resolver um sistema de equações lineares, enquanto o processo de treinamento da SVM consiste em resolver um problema de programação quadrática (*Quadratic Programming*, QP). Problemas de programação quadrática podem ser representados por uma função convexa, no qual é tão fácil de resolver quanto uma função linear, ou uma função não convexa, quando pode apresentar várias soluções viáveis. Com isso, apesar de resolver um sistema de equações lineares ser mais fácil que resolver um problema de programação quadrática, a ausência de esparsidade na solução, torna sua aplicação inconveniente a um grande conjunto de dados. A fim de superar tal desvantagem, neste trabalho são apresentadas duas novas abordagens às máquinas de vetores suporte, chamadas *Optimally Pruned LSSVM* (OP-LSSVC), quando aplicada a problemas de classificação e *Optimally Pruned LSSVR* (OP-LSSVR), quando aplicada a problemas de regressão. São apresentadas também outras duas abordagens semelhantes, chamadas *Matching Pursuit LSSVC* (MP-LSSVC), quando aplicada a problemas de classificação e *Matching Pursuit LSSVR* (MP-LSSVR), quando aplicada a problemas de regressão. A Máquina de Vetor Suporte de Mínimos Quadrados Otimamente Podada (*Optimally Pruned Least Squares Support Vector Machines*, OP-LSSVM), proposta deste trabalho, faz uso da Regressão Esparsa Multiresposta (*Multiresponse Sparse Regression*, MRSR), um método para obtenção de soluções esparsas, que durante o treinamento permite obter uma ordem dos vetores que mais representa uma variável dependente. Além disso, essa ordem fornecida pelo MRSR é reavaliada, de forma que um subconjunto de vetores suporte ainda melhor seja obtido. Para isso, o método de validação cruzada *Leave-one-out* (LOO) é utilizado. Essa proposta foi inspirada em uma recente metodologia, chamada OP-ELM (*Optimally Pruned Extreme Learning Machine*), a qual realiza a poda de neurônios da camada oculta em Máquinas de Aprendizagem Extrema (ELM). Neste trabalho, a LSSVM e o MRSR são utilizados de forma conjunta para obter um classificador e um regressor esparso. Por fim, pode-se concluir com os resultados obtidos que ambos os modelos, OP-LSSVM e OP-LSSVR, obtêm um desempenho equivalente ou superior em relação a outras metodologias aplicadas. De maneira semelhante, uma alternativa, utilizada para fins de comparação, nomeada como MP-LSSVM obtém um modelo esparso seguindo a mesma metodologia da OP-LSSVM, porém, utilizando um outro método de regressão linear esparsa, o *Matching Pursuit*.

Palavras-chave: Máquinas de Vetores Suporte, Máquinas de Vetores Suporte de Mínimos Quadrados, Solução Esparsa, Regressão Esparsa

Abstract

Least Square Support Vector Machines (LSSVMs) are an alternative to standard Support Vector Machines (SVM) because its training process, lagrange multipliers computation, is based on solving a linear equation system, while the training process for SVM relies on solving a quadratic programming optimization problem (QP). Quadratic programming problems can be represented by a convex function, which is as easy to solve as a linear function, or non-convex, when many viable solution can be available. By these means, despite solve a linear equations system is easier than solve a quadratic programming problem, the lack of sparsity in solution, and therefore in lagrange multipliers computation achieved after the LSSVM training, becomes a barrier when applied to a big set of data. In order to overcome this disadvantage, this works present two new support vector machines approaches, called Optimally Pruned LSSVM (OP-LSSVC), when applied to classification tasks and Optimally Pruned LSSVR (OP-LSSVR), when applied to regression tasks. Two others similar alternatives are shown, the Matching Pursuit LSSVC (MP-LSSVC), when applied to classification tasks and Matching Pursuit LSSVR (MP-LSSVR), when applied to regression tasks. The Optimally Pruned Least Support Vector Machines is based on Multiresponse Sparse Regression (MRSR), a method to obtain sparse solution that while training is possible to get the order of vectors that most represent a dependent variable. In addition, the aforementioned order is reassessed, such that an even better subset could be achieved. For that, the Leave-one-out cross validation method (LOO) is used. This approach was inspired by a recent methodology called OP-ELM, which prunes hidden neurons of Extreme Learning Machines. Therefore, in this paper, we put LSSVM and MRSR to work together in order to achieve sparse classifiers and regressors, as well as one can see that we achieved equivalent or even superior performance for real-world classification and regression tasks with respect to other methodologies applied in this work. Lastly, we can conclude with obtained results the both models, OP-LSSVM and OP-LSSVR, achieve a equivalente or superior performance with respect to other applied methodologies. Similarly, the MP-LSSVM achieve a sparse model adopting the same methodology of OP-LSSVM but, using other sparse linear regression method.

Key-words: Support Vector Machines, Least Squares Support Vector Machines, Sparse Solution, Sparse Regression

Lista de ilustrações

- Figura 1 – Hiperplano de decisão. A linha preta representa o hiperplano que melhor separa as classes. Em cinza as margens equidistantes do hiperplano. . . 25
- Figura 2 – Interpretação geométrica para o cálculo das margens da SVM. a) Projeção de $\mathbf{x}_i$ sobre o hiperplano de decisão. b) Margem de separação ρ . A margem de separação é exposta na cor cinza. $r^{(-s)} = r^{(+s)}$ uma vez que ambos os vetores suporte $\mathbf{x}_1^{(s)}$ e $\mathbf{x}_2^{(s)}$ são equidistantes ao hiperplano de decisão..... 27
- Figura 3 – Interpretação da SVM com margens flexíveis. Perceba que ao permitir que alguns padrões sejam classificados de forma errada, como por exemplo o ponto azul mais inferior, é possível escolher um hiperplano que classifique corretamente a grande maioria dos pontos. 29
- Figura 4 – A linha contínua na Figura a) é apresenta, a função estimada pelo s-SVR. A linha pontilhada define as margens correspondentes a um erro de $\pm s$. Na Figura b) é apresentada a forma da função proposta por Vapnik..... 31
- Figura 5 – Diversos valores de s para um mesmo conjunto de dados. Na Figura a) e b) perceba que o valor de s é maior que o necessário para criar um tubo adequado ao redor dos dados, permitindo que tal tubo se posicione de várias formas sem acompanhar a dinâmica dos dados. Na Figura c) um valor pequeno de s torna o problema sem solução uma vez que algumas das restrições serão sempre violadas..... 32
- Figura 6 – Resultado obtido com a flexibilização da margem do tubo. a) A reta em preto, não pontilhada, não está bem ajustada aos pontos mesmo assim não viola as restrições apresentadas na Equação 2.27. b) Ao ignorar alguns dos pontos obtem-se uma função melhor ajustada. 33
- Figura 7 – Fluxograma do treinando da OP-LSSVM. 51
- Figura 8 – Resultados das simulações utilizando os 3 primeiros conjuntos artificiais apresentados na Tabela 2. Do lado esquerdo: os pontos azuis são os dados de treinamento. Em vermelho os vetores suporte utilizados pela OP-LSSVR para estimar a curva em verde. Em preto a curva estimada pela LSSVR. Do lado direito: erro obtido em função da quantidade de vetores adicionados para estimar a curva à esquerda 64

- Figura 9 – Resultados das simulações utilizando os 3 últimos conjuntos artificiais apresentados na Tabela 2. Do lado esquerdo: os pontos azuis são os dados de treinamento. Em vermelho os vetores suporte utilizados pela OP-LSSVR para estimar a curva em verde. Em preto a curva estimada pela LSSVR. Do lado direito: erro obtido em função da quantidade de vetores adicionados para estimar a curva à esquerda65
- Figura 10 – Resultados das simulações utilizando os 4 conjuntos artificiais de classificação. Do lado esquerdo: os pontos azuis e vermelho são os dados de treinamento. Em verde, ao centro, os vetores suporte utilizados pela OP-LSSVC para estimar a região de decisão. Do lado direito: erro obtido em função da quantidade de vetores adicionados para estimar a função discriminante.....73

Lista de tabelas

Tabela 1 – Funções de <i>kernel</i> tipicamente utilizadas em máquinas de vetores suporte.	35
Tabela 2 – Descrição dos problemas artificiais. $rand_i$ é uma amostra aleatória do ruído.	62
Tabela 3 – Descrição dos conjuntos de dados reais utilizados para regressão.	66
Tabela 4 – Resultado <i>kernel</i> Linear Parte 1. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> linear. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.	67
Tabela 5 – Resultado <i>kernel</i> Linear Parte 2. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> linear. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.	68
Tabela 6 – Resultado <i>kernel</i> RBF Parte 1. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> RBF. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.	69
Tabela 7 – Resultado <i>kernel</i> RBF Parte 2. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> RBF. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.	70
Tabela 8 – Percentual de redução entre o regressor apresentado em cada linha e a LSSVR com <i>kernel</i> linear.	71
Tabela 9 – Percentual de redução entre o regressor apresentado em cada linha e a LSSVR com <i>kernel</i> RBF.	71
Tabela 10 – Descrição dos conjuntos de dados reais utilizados para classificação. . .	74

Tabela 11 – Resultado <i>kernel</i> Linear Parte 1 - Acurácia e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> Linear. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.	76
Tabela 12 – Resultado <i>kernel</i> Linear Parte 2 - Acurácia e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> Linear. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.	77
Tabela 13 – Resultado <i>kernel</i> RBF Parte 1 - Acurácia e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> RBF. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.	78
Tabela 14 – Resultado <i>kernel</i> RBF Parte 2 - Acurácia e desvio padrão após 30 execuções independentes utilizando <i>kernel</i> RBF. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.	79
Tabela 15 – Percentual de redução entre o classificador apresentado em cada linha e a LSSVM com <i>kernel</i> linear.	80
Tabela 16 – Percentual de redução entre o regressor apresentado em cada linha e a LSSVM com <i>kernel</i> RBF.	80

Lista de abreviaturas e siglas

BE	Backward Elimination (Eliminação para Trás)
BS	Backward Selection (Seleção para Trás)
BSS	Best Subset Selection (Seleção de Melhor Subconjunto)
ELM	Extreme Learning Machine (Máquinas de Aprendizado Extremo)
EQM	Erro quadrático médio
FS	Forward Selection (Seleção Progressiva)
FSS	Forward Stagewise Selection (Seleção Progressiva em Estágios)
LARS	Least Angle Regression
LOO	Leave-one-out
LSSVC	Least Squares Support Vector Classifier (Classificador de Vetor Suporte de Mínimos Quadrados)
LSSVM	Least Squares Support Vector Machines (Máquina de Vetor Suporte de Mínimos Quadrados)
LSSVR	Least Squares Support Vector Regressor (Regressor de Vetor Suporte de Mínimos Quadrados)
MLP	Multilayer Perceptron
MP	Matching Pursuit
MP-LSSVM	Mathing Pursuit LSSVM
MRSR	Multiresponse Sparse Regression
OMP	Orthogonal Matching Pursuit
OP-ELM	Optimally Prunned Extreme Learning Machine
OP-LSSVC	Optimally Prunned LSSVM (Classificador de Vetor Suporte Otimamente Podada)
OP-LSSVM	Optimally Prunned LSSVM (Máquina de Vetor Suporte Otimamente Podada)

OP-LSSVR	Optimally Prunned LSSVM (Regressor de Vetor Suporte Otimamente Podada)
QP	Quadratic Problem
QP-SVM	Quadratic Programming SVM
PRESS	Prediction Sum of Squares Error (Predição da Soma dos Erros Quadradados)
RBF	Radial Basis Function
SMO	Sequential Minimal Optimization
SVC	Support Vector Classifier (Classificador de Vetor Suporte)
SVD	Single Value Decomposition
SVM	Support Vector Machines (Máquina de Vetor Suporte)
SVR	Support Vector Regressor (Regressor de Vetor Suporte)

Lista de símbolos

N	Quantidade total de padrões
p	Dimensão do padrão $\mathbf{x}_i$
$\mathbf{w}$	Vetor de pesos
b	Intercepto
$\mathbf{x}$	Vetor de atributos ou características
$\mathbf{x}_i$	i-ésimo vetor de atributos
y_i	Rótulo do i-ésimo padrão
$\mathbf{w}_o$	Vetor de pesos ótimo
b_o	Intercepto ótimo
$\mathbf{x}^{(s)}$	Vetor suporte
$\mathbf{x}_p$	Projeção do vetor $\mathbf{x}$ sobre o hiperplano de decisão
$r^{(s)}$	Distância de um vetor suporte ao hiperplano de decisão
ρ	Largura da margem de separação entre classes
$J(\mathbf{w})$	Função custo em função de $\mathbf{w}$
a_i	i-ésimo multiplicador de Lagrange
$\mathbf{a}$	Vetor de multiplicadores de Lagrange
ε_i	Variável de folga associada ao padrão $\mathbf{x}_i$
γ	Fator de regularização
σ	Fator de abertura de uma função RBF
$\boldsymbol{\beta}$	Multiplicadores de Lagrange associados às variáveis de folga
s	Erro na estimativa da função de regressão
ξ_i	Variáveis de folga na SVR
ξ_i^*	Complemento das variáveis de folga na SVR

α^*	Complemento dos multiplicadores de Lagrange na SVR
η	Segundo conjunto de multiplicadores de Lagrange da SVR
η^*	Complemento do segundo conjunto de multiplicadores de Lagrange da SVR
$L(., ., .)$	Lagrangeano de uma função com três variáveis
$L(., ., ., .)$	Lagrangeano de uma função com quatro variáveis
Ω	Resultado da aplicação da função de <i>kernel</i> na formulação do SVC
Φ	Resultado da aplicação da função de <i>kernel</i> na formulação do SVR
τ	Percentual que será removido em cada passo da P-LSSVM
c_j^k	Correlação de $\mathbf{x}_j$ com o resíduo na iteração k
$\mathbf{t}$	Vetor de rótulo do <i>kernel</i> utilizada para estimar os multiplicadores de Lagrange
$\mathbf{D}$	Matriz de <i>kernel</i> utilizada para estimar os multiplicadores de Lagrange
$\mathbf{D}_*$	Matriz de <i>kernel</i> reduzida utilizada para estimar os multiplicadores de Lagrange
$\mathbf{d}_i$	i -ésima vetor coluna da matriz $\mathbf{D}$
$\mathbf{r}_j$	j -ésima vetor linha da matriz $\mathbf{D}$
$\mathbf{z}$	Solução do sistema linear $\mathbf{D}\mathbf{z} = \mathbf{t}$
A	Conjunto de índices das variáveis que podem ser escolhidas do modelo
U	Conjunto dos índices dos padrões em uso
δ_i	Resíduo da estimativa do modelo
S_p	Conjunto dos 2^p vetores utilizados no MRSR
B	Passo de treinamento para as técnicas de seleção progressiva
V	Conjunto com índices dos vetores ordenados pelo MRSR
V^*	Subconjunto ótimo dos índices dos vetores suporte
W_k	Subconjunto a ser testado pelo <i>leave-one-out</i>

Sumário

I	INTRODUÇÃO	18
1.1	Motivação	19
1.2	Objetivos Geral e Específicos	21
1.2.1	Objetivo Geral	21
1.2.2	Objetivos Específicos	22
1.3	Produção Científica	22
1.4	Estrutura da Dissertação	22
2	MÁQUINAS DE VETORES SUPORTE	24
2.1	Introdução	24
2.2	Máquinas de Vetores Suporte para Classificação	24
2.2.1	SVM de Margens rígidas	25
2.2.2	SVM - Margem flexível	28
2.3	Máquinas de Vetores Suporte para Regressão	30
2.4	Truque de <i>Kernel</i>	35
2.5	Máquinas de Vetores Suporte de Mínimos Quadrados para Classifi- cação	36
2.6	Máquinas de Vetores Suporte de Mínimos Quadrados para Regressão	38
2.7	Máquinas de Vetores Suporte Esparsas	39
2.7.1	P-LSSVM	39
2.7.2	IP-LSSVM	41
2.8	Conclusão	42
3	REDUÇÃO DE MODELOS LINEARES	44
3.1	Introdução	44
3.2	Seleção de Modelos	45
3.2.1	Seleção de Melhor Subconjunto	46
3.2.2	Seleção Progressiva	46
3.2.3	Seleção Progressiva em Estágios	48
3.2.4	Eliminação para Trás	50
3.3	Máquinas de Vetores Suporte de Mínimos Quadráticos Otimamente Podadas (Proposta)	50
3.3.1	Seleção através da Regressão Esparsa Multiresposta	52
3.3.2	Predição da Soma dos Erros Quadrados (PRESS)	54
3.3.3	Construção do Modelo Final	57
3.3.4	Conclusão	59

4	RESULTADOS E DISCUSSÃO	60
4.1	Escolha dos parâmetros do modelo	60
4.2	Avaliação do desempenho	61
4.3	Resultados Obtidos em Problemas de Regressão	62
4.3.1	Conjunto de dados artificiais	62
4.3.2	Conjunto de dados reais	63
4.4	Resultados Obtidos em Problemas de Classificação	72
4.4.1	Conjunto de dados artificiais	72
4.4.2	Conjunto de dados reais	74
5	CONCLUSÃO E TRABALHOS FUTUROS	81
	REFERÊNCIAS	82
	APÊNDICES	88
	APÊNDICE A – EQUIVALÊNCIA ENTRE O PRESS E O LEAVE- ONE- OUT	89

1 Introdução

Existe um grande conjunto de técnicas estatísticas que tem por finalidade resolver problemas de classificação binária, dentre as mais conhecidas e comumente utilizadas estão a análise discriminante, regressão logística e modelos estatísticos não paramétricos, a exemplo das Redes Neurais Artificiais (RNAs). Por volta dos anos 30, Fisher (1936) sugeriu o primeiro algoritmo para reconhecimento de padrões que pode ser definido como o meio no qual um padrão ou sinal de entrada é classificado (rotulado) como integrante de uma classe. Dado um problema com duas classes, Fisher assumiu que cada classe possui uma distribuição normal em que os parâmetros, média e variância são obtidos a partir de um conjunto de treinamento. A partir dos quais, a função discriminante pode ser obtida para então realizar a tomada de decisão (ANCHALIA; ROY, 2015).

Nos anos 50, Rosenblatt (1958) estudou outras técnicas de aprendizagem de máquina, tais como, os perceptrons e as Redes Neurais Artificiais (RNA). Rosenblatt, porém, não definiu um algoritmo de treinamento que ajustasse todos os pesos dos neurônios de uma RNA como por exemplo, os pesos dos neurônios ocultos. No entanto, em seu trabalho foi proposto uma técnica adaptativa para obtenção dos pesos da camada de saída.

Ao se considerar o projeto de um classificador, o objetivo principal durante o seu desenvolvimento é projetá-lo de forma que se obtenha um bom desempenho quando este é apresentado a dados não utilizados durante o treinamento. Uma vez que o conjunto de treinamento é suficientemente representativo, tanto em quantidade quanto em diversidade, pode-se dizer que tem-se o necessário para contruir um classificador robusto. Porém, geralmente, o conjunto de treinamento é muito pequeno, o que pode levar a uma perigosa situação, na qual se faz uma estimativa de um valor da uma função onde não há dados de treinamento próximo ao ponto estimado (NAGY, 1968).

Uma Máquina de Vetor Suporte (*Support Vector Machine*, SVM) é um método de aprendizado supervisionado desenvolvido por Vladimir Vapnik (1995) nos anos 90. Desde sua criação, seu desempenho, em geral, se mostra superior aos demais métodos, tanto em tarefas de classificação quanto de regressão. As máquinas de vetores suporte são técnicas não paramétricas de classificação ou regressão que já mostraram ótimos resultados na área de diagnósticos médicos, reconhecimento de padrões, previsão do tempo entre outras tarefas similares (AURIA, 2008). De fato, os recursos matemáticos, oriundos da área de otimização e aprendizagem estatístico colocou a SVM em uma posição de destaque na última década.

Máquinas de vetores suporte têm sido amplamente utilizadas devido à sua capacidade de generalização, e como dito, geralmente alcançam melhores resultados do que outros

métodos com dados não vistos durante o treinamento. Essa vantagem, em relação a outros métodos, é devida em grande parte a preocupação na redução conjunta do risco empírico e do risco estrutural. O primeiro refere-se aos erros durante a etapa de treinamento e o segundo aos erros com relação à padrões não vistos durante essa etapa.

1.1 Motivação

O processo de treinamento das SVMs é um problema de otimização quadrática (*Quadratic Optimization Problem*, QP) que exige o armazenamento de uma matriz quadrada de dimensões iguais ao número de padrões, devido a essa necessidade, em geral, torna-se inviável encontrar a solução em conjuntos de treinamento com grande quantidade de amostras. O *Sequential Minimal Optimization* (SMO) é um algoritmo que reduz o problema de otimização como um todo ao dividi-lo em pequenos problemas de otimização, que são resolvidos analiticamente e têm o custo computacional reduzido (PLATT, 1999).

As máquinas de vetores suporte de mínimos quadrados (*Least Squares Support Vector Machines*, LSSVM) são uma alternativa às SVMs, pois o treinamento destes modelos consistem em resolver um sistema linear ao invés de exigir a resolução de um problema de programação quadrática (SUYKENS; VANDEWALLE, 1999). De uma maneira geral, as LSSVMs fazem uso de uma inversão de matriz ¹ para resolver o sistema linear, o que reduz consideravelmente o tempo de treinamento.

A redução do problema de QP a um problema linear, porém, traz uma desvantagem. Enquanto as LSSVMs fazem uso de todos os padrões de treinamento para obter o modelo final e estimar a saída de um padrão apresentado, as SVMs selecionam apenas alguns destes padrões, denominados vetores suporte. Devido a esta utilização, o modelo da LSSVM é muito maior que o modelo estimado pela SVM. Em outras palavras, o modelo obtido pela LSSVM não é esparso.

A comunidade acadêmica desenvolveu pesquisas significantes para acelerar e simplificar o treinamento das LSSVMs. Mesmo que, de maneira geral, o motivo destas seja a redução do custo computacional da SVM, algumas técnicas também trazem modelos mais compactos em termos de vetores suporte. Grande parte dos métodos tentam realizar a redução do número de vetores suporte apenas para a fase de teste, dessa forma todos os dados são utilizados durante o treinamento do modelo, como, por exemplo, o método apresentado por Burges (1996). Outros pesquisadores obtêm uma quantidade menor de vetores suporte também durante o treinamento, reduzindo assim o custo computacional para a obtenção da superfície de decisão (LU; MENG; CAO, 2010; KOGGALAGE; HALGAMUGE, 2004).

¹ Com a aplicação da Decomposição em valores Singulares, (*Single Value Decomposition*, SVD) é possível obter resultados numericamente mais estáveis (LEE, 2012).

Pode-se distinguir as principais abordagens ao problema de esparsidade em redução antes e após o treinamento. Para o primeiro caso, o algoritmo de treinamento faz uma seleção que minimiza o número de vetores suporte de maneira direta, mantendo uma pequena quantidade destes durante a fase de treinamento. Para o último, o modelo da LSSVM contendo todos os dados como vetores suporte são removidos de forma iterativa, geralmente, com um alto custo computacional nas fases iniciais, portanto sendo o menos recomendável para um grande volume de dados. Uma vez que intenção é desenvolver um modelo de treinamento que possa ser utilizado em um grande volume de dados, a proposta deste trabalho aborda técnicas que se encaixam na primeira categoria.

Na primeira categoria tem-se, por exemplo, as máquinas de vetores suporte de tamanho fixo (*Fixed-size least squares support vector machines*, FS-LSSVM), abordada por Suykens T. Van Gestel (2002) esta técnica faz uma aproximação do espaço de características utilizando o método de Nyström (PRENTER, 1981), o qual aproxima a matriz de *kernel* à uma de menor ordem. Uma versão utilizando validação cruzada, mostrou que o erro da FS-LSSVM diminui ao passo que os vetores suporte são adicionados e então este estabiliza em um valor mínimo. O erro inicial e a velocidade de convergência são fortemente influenciadas pela seleção dos vetores iniciais, essa seleção foi feita utilizando a entropia de Rènyi (BRABANTER *et al.*, 2010). Em (MEHRKANOON *et al.*, 2015), outro trabalho recente e ainda sob co-autoria de Suykens, foca em viabilizar o treinamento para um grande número de dados.

As máquinas de vetores suporte de conjunto reduzido (*Reduced Set Support Vector Machine*, RS-SVM), publicado por Hussain *et al.* (2008), realiza uma aproximação do *kernel* $K(\mathbf{x}, \mathbf{x}_j)$ utilizando um conjunto reduzido de vetores suporte, estes escolhidos utilizando as regras publicadas por Burges (1996) e Scholkopf *et al.* (1999). Tange e Mazzoni (2006) apresentaram uma versão multiclasse também baseada nas simplificações de Burges. Downs, Gates e Masters (2002) e Liang (2010), mostraram alternativas ao identificar vetores linearmente dependentes dentro do conjunto de treinamento (LAY, 2002). Em outra alternativa, a poda dos vetores suporte é realizada com base na distância destes para a região de decisão. Em cada iteração do algoritmo uma parcela é removida enquanto não obedecer um critério de parada, (SUYKENS; LUKAS; VANDEWALLE, 2000b; SUYKENS; LUKAS; VANDEWALLE, 2000a).

Na segunda categoria, o número de vetores suporte utilizados no modelo são escolhidos após o treinamento. Carvalho e Braga (2009) propõe um método de seleção baseado nos valores dos multiplicadores de Lagrange. Carvalho assume que multiplicadores de Lagrange com valores muito maiores que zero são mais prováveis de se tornarem um vetor suporte, dessa forma, realiza a remoção dos vetores suporte com os multiplicadores mais próximos de zero. Geebelen, Suykens e Vandewalle (2012) obtém um solução aproximada da SVM fazendo uma série de suposições a fim de obter um problema de classificação

separável no espaço de características. Para isso, o treinamento é realizado com todos os dados, e em seguida são removidas todas as amostras de treinamento que são classificadas de forma errada. A superfície de decisão obtida é similar a primeira, porém, o modelo final contém uma quantidade menor de vetores suporte.

Keerthi, Chapelle e DeCoste (2006) realizam uma busca extensiva, de maneira gulosa, por todas as amostras de treinamento. A seleção do próximo vetor suporte a ser incluído é feita em função do erro máximo obtido pelo modelo, escolhendo o vetor que o minimiza. Devido à essa busca, a proposta de Keerthi só é aplicável a um conjunto já reduzido de dados. Seguindo a mesma ideia da minimização do erro obtido outras técnicas, envolvendo meta-heurísticas, como algoritmos genéticos (SILVA; NETO, 2013) e têmpera simulada (ZHANG; NIU, 2011), também podem ser citadas.

Um modelo esparso permite que após o treinamento seja utilizada poucas informações para estimar a classe de algum padrão. Na prática a utilização de métodos diferentes para se treinar SVMs pode obter quantidades diferentes de vetores suporte, mesmo que aplicado ao mesmo conjunto de treinamento (D'AMATO; MORENO; MUJICA, 2004).

Em (KARSMAKERS *et al.*, 2011) os autores utilizam a formulação do problema primal apresentada por Suykens T. Van Gestel (2002) e realizam uma aproximação esparsa do *kernel*. Essa aproximação é feita utilizando técnicas de seleção de modelos lineares como: *Matching Pursuit*, *Least Angle Regression* e o *Sparse Conjugate Direction*. Uma vez que a obtenção dos multiplicadores de Lagrange das LSSVMs consiste na solução de um sistema linear e, pode-se utilizar técnicas de regressão linear esparsa a fim de obter um modelo reduzido. O uso destas técnicas torna alguns dos multiplicadores de Lagrange, obtidos através da solução do sistema linear, iguais a zero e, portanto, os vetores suporte associados a estes podem ser removidos. A proposta deste trabalho, chamada OP-LSSVM, de maneira semelhante a seguida por Karsmakers *et al.* (2011), realiza uma aproximação do *kernel* LSSVM utilizando técnicas de regressão esparsa, porém no problema dual. Além disso, é incluída uma fase a mais responsável por obter um modelo ainda menor, isso é feito através da avaliação dos vetores suporte escolhidos pela técnica de regressão.

1.2 Objetivos Geral e Específicos

1.2.1 Objetivo Geral

O Objetivo deste trabalho é apresentar alternativas para a obtenção de um modelo esparso para as máquinas de vetores suporte baseadas em mínimos quadrados quando aplicadas a problemas de regressão ou classificação, mantendo o erro estimado equiparável às técnicas atuais.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, têm-se os seguintes objetivos específicos.

- Avaliar o desempenho em relação a tempo, capacidade de generalização e tamanho do modelo final, de classificadores e regressores SVM e LSSVM, quando utilizados com diferentes tipos de *kernel*.
- Apresentar uma nova alternativa para o treinamento da LSSVM com base em um algoritmo de regressão linear esparsa.
- Avaliar o desempenho da nova alternativa em relação a outros métodos esparsos de classificação e regressão baseados na LSSVM.

1.3 Produção Científica

Durante o decorrer deste projeto de mestrado, foram publicados os seguintes trabalhos científicos:

- **VIEIRA, D. C. S. ; ROCHA NETO, A. ; RODRIGUES, A. W. O.**, Sparse Least Squares Support Vector Regression via Multiresponse Sparse Regression. In: IEEE World Congress on Computational Intelligence (IEEE WCCI), 2016, Vancouver. International Joint Conference on Neural Networks (IJCNN 2016), 2016.
- **VIEIRA, D. C. S. ; ROCHA NETO, A. ; RODRIGUES, A. W. O.**, Sparse Least Squares Support Vector Machines via Multiresponse Sparse Regression. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2016, Bruges.
- Lopes, G. S.; **VIEIRA, D. C. S.**; RODRIGUES, A. W. O.; Rebouças Filho, P.P., Recognition of handwritten digits using the signature features and Optimum-Path Forest Classifier. IEEE Latin America Transactions, vol. 14, no. 5, pp. 2455-2460, Maio 2016.

1.4 Estrutura da Dissertação

Após o capítulo de introdução, onde é apresentada uma breve revisão bibliográfica de máquinas de vetores suporte de mínimos quadrados esparsas, é abordada a fundamentação teórica para a proposta deste trabalho. No Capítulo 2, são apresentadas uma abordagem de classificadores SVM e o conceito do hiperplano de separação entre as classes e sua fundamental importância na obtenção da generalização em classificadores, este permitindo geralmente resultados mais precisos e menores taxas de erros à padrões não apresentados

durante o treinamento. É apresentada também uma abordagem teórica aos regressores de vetores suporte. Ainda neste capítulo, as máquinas de vetores suporte de mínimos quadrados para classificação, LSSVC, e sua versão para regressão, LSSVR, são expostos, exibindo os passos necessários para obtê-las a partir da formulação padrão das máquinas de vetores suporte. Ao fim do capítulo são apresentados dois métodos para a obtenção de esparsidade em Máquinas de Vetores Suporte de Mínimos Quadrados. São ressaltados suas características e meios de obtenção do modelo esparsos. São eles, P-LSSVM e IP-LSSVM.

No Capítulo 3 são expostas técnicas para a obtenção de modelos lineares reduzidos. De maneira geral, serão apresentadas técnicas gulosas de seleção, mas que possuem características de baixo custo computacional. São estas: *Best Subset Selection*, *Forward Selection*, *Forward Stagewise Selection*, *Backward Elimination* e o *Multiresponse Regression*. Esta última é utilizada na proposta deste trabalho. No Seção 3.2 a proposta é apresentada. Na Subseção 3.2.1 é apresentada, em detalhes, como se dá a obtenção de um modelo esparsos através da técnica MRSR. Nas duas subseções seguintes é discutida uma técnica de validação cruzada que será utilizada junto com a proposta deste trabalho. Essa técnica visa a escolha do melhor subconjunto de variáveis a ser utilizada por um modelo de regressão linear. Finalmente na Subseção 3.2.4 são apresentados os passos necessários para a construção do modelo proposto.

No Capítulo 4 são apresentados os resultados obtidos através de simulações em suas versões de classificação e regressão com as seguintes máquinas de vetores suporte: SVM, LSSVM, P-LSSVM, IP-LSSVM, OP-LSSVM e MP-LSSVM, sendo que, as duas últimas mencionadas são a proposta deste trabalho e uma alternativa semelhante que estende um método já existente. Neste capítulo são apresentadas simulações com bases artificiais, as quais permitem uma visualização das curvas estimadas para regressão e da região de decisão traçadas pela OP-LSSVM e pela LSSVM. São apresentadas também simulações com bases reais, as quais permitem uma verificação do tempo de treinamento, compactação do modelo e desempenho em termos de acurácia e erro quadrático médio.

No Capítulo 5, apresentadas a conclusão e trabalhos futuros, onde são ressaltadas a introdução de outros métodos a fim de obter modelos esparsos e uma técnica para otimizar o cálculo de operações matriciais. Por fim, é apresentado no apêndice a prova de equivalência entre o *leave-one-out* e o PRESS, o qual é uma técnica de validação cruzada utilizada junto com a proposta.

2 Máquinas de Vetores suporte

2.1 Introdução

Neste capítulo são apresentados os métodos que visam impor esparsidade em máquinas de vetores suporte de mínimos quadrados. Inicialmente o problema de otimização para a SVM será abordado e logo após, as novas condições de optimalidade que foram aplicadas para o desenvolvimento da LSSVM. As SVMs tornaram-se populares por serem capazes de resolver problemas de classificação e regressão e por se preocuparem com padrões não vistos durante o treinamento. Nesta seção também é discutida a formulação matemática e os critérios de otimização para realizar o treinamento da SVM para classificação.

2.2 Máquinas de Vetores Suporte para Classificação

Considere a seguinte função linear que representa um hiperplano e que pode ser utilizada para resolver problemas de classificadores de padrões dado por

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \quad (2.1)$$

em que $\mathbf{w}$ é o vetor de pesos, b é o viés e $\mathbf{x}$ é um padrão que se deseja estimar a classe. Durante a fase de treinamento o vetor $\mathbf{w}$ e o intercepto b são estimados, considerando um problema de classificação onde as classes são linearmente separáveis, existem infinitos hiperplanos que permitem essa separação. A Figura 1 representa alguns dos possíveis hiperplanos que podem ser estimados a partir de um conjunto de dados linearmente separável. Uma simples análise dos possíveis hiperplanos que permitem essa separação indica que o hiperplano representado pela linha preta é o ideal, uma vez que este permite uma maior separação entre as classes.

A SVM, diferentemente de outros classificadores como o MLP, ELM, RBF entre outros, tem uma preocupação maior com o hiperplano de decisão, essa preocupação se reflete na maximização da margem de separação entre as classes. A margem é definida como duas vezes a distância perpendicular entre um padrão $\mathbf{x}_i$ e o hiperplano de decisão.

Serão aqui abordados dois modelos, SVM margem rígida onde só pode ser aplicado em problemas linearmente separáveis e a SVM margem flexível que pode ser aplicado de maneira geral em qualquer tipo de problema de classificação.

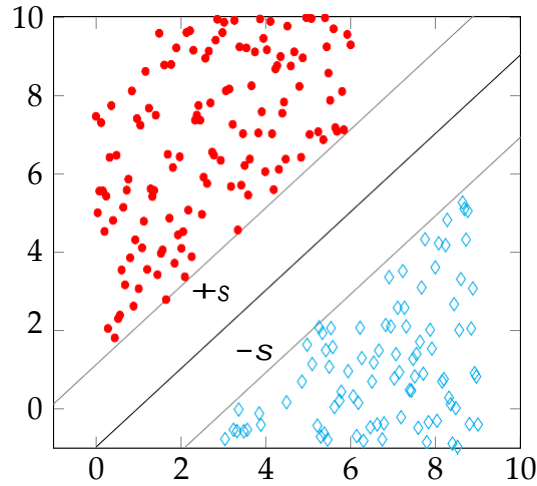


Figura 1 – Hiperplano de decisão. A linha preta representa o hiperplano que melhor separa as classes. Em cinza as margens equidistantes do hiperplano.

2.2.1 SVM de Margens rígidas

Considere um conjunto de treinamento linearmente separável $\{\mathbf{x}_i, y_i\}_{i=1}^N$, em que $\mathbf{x}_i \in \mathbb{R}^p$ são os vetores de entrada e $y_i \in \{-1, 1\}$ são os rótulos das classes negativa e positiva respectivamente.

A escolha da classe de um padrão $\mathbf{x}$ é feita através da função discriminante f :

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}_o^T \mathbf{x} + b_o) \quad (2.2)$$

Em que $\mathbf{w}_o$ e b_o é o vetor de pesos e o intercepto ótimo, respectivamente, estimados pela SVM durante o treinamento. Uma vez que se considera o conjunto de dados de treinamento como linearmente separável pode-se apresentar as seguintes inequações:

$$\begin{aligned} \exists \mathbf{w}_o^T \mathbf{x}_i + b_o &\geq 1, & \text{se } y_i = +1 \\ \exists \mathbf{w}_o^T \mathbf{x}_i + b_o &\leq -1, & \text{se } y_i = -1 \end{aligned} \quad (2.3)$$

De uma forma mais concisa pode-se representar as inequações por

$$y_i[\mathbf{w}_o^T \mathbf{x}_i + b_o] \geq 1. \quad (2.4)$$

Devido a formulação das restrições apresentadas na Equação (2.3), cada inequação define uma das margens de separação, com isso, os padrões que estão sobre a margem são chamados de vetores suporte $\mathbf{x}^{(s)}$. $f(\mathbf{x}^{(s)})$ podem assumir valores -1 ou +1 de acordo com a classe que $\mathbf{x}^{(s)}$ pertence

Para estimar a largura da margem de separação entre as classes, considere o vetor $\mathbf{x}_p$ como sendo a projeção do vetor $\mathbf{x}$ sobre o hiperplano de decisão. Neste contexto pode-se assim reescrever $\mathbf{x}$ da seguinte forma:

$$\mathbf{x} = \mathbf{x}_p + r \frac{\mathbf{w}_o}{\|\mathbf{w}_o\|}. \quad (2.5)$$

Uma vez que $\mathbf{x}_p$ está sobre o hiperplano de decisão, sabe-se que $f(\mathbf{x}_p) = 0$. Levando isto em consideração, pode-se reescrever a função de decisão para os parâmetros ótimos ($\mathbf{w}_o$ e b_o) da seguinte forma:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{w}_o^T \mathbf{x} + b_o \\ f(\mathbf{x}) &= \mathbf{w}_o^T \mathbf{x}_p + r \frac{\mathbf{w}_o^T \mathbf{w}_o}{\|\mathbf{w}_o\|} \\ f(\mathbf{x}) &= \underbrace{\mathbf{w}_o^T \mathbf{x}_p}_{f(\mathbf{x}_p)=0} + \underbrace{b_o}_{f(\mathbf{x}_p)=0} + \mathbf{w}_o^T r \frac{\mathbf{w}_o}{\|\mathbf{w}_o\|} \end{aligned} \quad (2.6)$$

e, como $\|\mathbf{w}_o\| = \sqrt{\mathbf{w}_o^T \mathbf{w}_o}$ tem-se que $f(\mathbf{x}) = r \frac{\mathbf{w}_o^T \mathbf{w}_o}{\|\mathbf{w}_o\|}$ resultando em $f(\mathbf{x}) = r \|\mathbf{w}_o\|$ e por fim $r = \frac{f(\mathbf{x})}{\|\mathbf{w}_o\|}$ em que r é um escalar que torna a Equação (2.5) verdadeira.

Finalmente pode-se concluir que a margem de separação entre as classes $\rho = 2r^{(s)}$, em que $r^{(s)}$ é a distância de um vetor suporte $\mathbf{x}^{(s)}$ para o hiperplano de decisão, e sabendo que $f(\mathbf{x}^{(s)}) = 1$ uma vez que $\mathbf{x}^{(s)}$ está sobre a margem de separação, a margem pode ser obtida por:

$$\begin{aligned} \rho &= 2r^{(s)} \\ \rho &= 2 \frac{f(\mathbf{x}^{(s)})}{\|\mathbf{w}_o\|} \\ \rho &= 2 \frac{1}{\|\mathbf{w}_o\|} \end{aligned} \quad (2.7)$$

Na Figura 2 é apresentada a interpretação geométrica do escalar r , da margem de separação entre as classes ρ , dos hiperplanos ótimos e os de suporte.

Como dito anteriormente, o processo de treinamento das SVMs consiste em obter um hiperplano que maximiza a margem de separação entre as classes. Uma vez que a largura da margem pode ser definida de acordo com a Equação (2.7), pode-se inferir que maximizar ρ é equivalente a minimizar a norma de $\mathbf{w}$ ou, de maneira equivalente, minimizar $J = \frac{1}{2} \mathbf{w}^T \mathbf{w}$. Além disso, os dados de treinamento devem ser corretamente classificados, e portanto, a Equação (2.4) deve ser sempre satisfeita. Do exposto, se define o problema primal de otimização para máquinas de vetores suporte para problemas de classificação de padrões, a saber:

$$\square \text{Minimizar } J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (2.8)$$

$$\square \text{Sujeito a } y_i [\mathbf{w}_o^T \mathbf{x}_i + b_o] - 1 \geq 0$$

Este problema de otimização, aqui sujeito a restrições lineares, pode ser resolvido através do método de Lagrange que permite representar o problema de otimização descrito na equação anterior por:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i (y_i [\mathbf{w}^T \mathbf{x}_i + b] - 1), \quad (2.9)$$

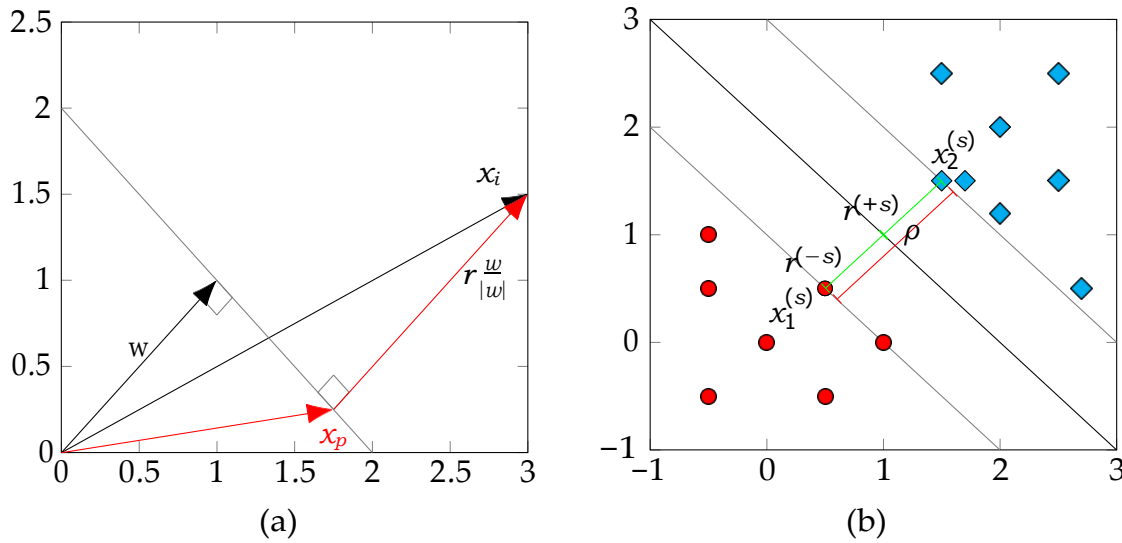


Figura 2 – Interpretação geométrica para o cálculo das margens da SVM. a) Projeção de $\mathbf{x}_i$ sobre o hiperplano de decisão. b) Margem de separação ρ . A margem de separação é exposta na cor cinza. $r^{(-s)} = r^{(+s)}$ uma vez que ambos os vetores suporte $\mathbf{x}_1^{(s)}$ e $\mathbf{x}_2^{(s)}$ são equidistantes ao hiperplano de decisão

em que a_i são os multiplicadores de Lagrange, ou seja $a_i \geq 0 \forall i$. A solução deste problema consiste em encontrar o ponto de sela da função descrita na Equação (2.9) em que se busca a maximização de α e a minimização de $\mathbf{w}$ e b , ou seja,

$$\max_{\alpha} \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha). \quad (2.10)$$

Considere então as condições de optimalidade para $L(\mathbf{w}, b, \alpha)$

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} &= 0 \rightarrow \mathbf{w} = \sum_{i=1}^N a_i y_i \mathbf{x}_i \\ \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} &= 0 \rightarrow \sum_{i=1}^N a_i y_i = 0 \end{aligned} \quad (2.11)$$

Após algumas manipulações algébricas, apresentadas na Equação (2.12), e ao substituir a equação atribuída a $\mathbf{w}$ da Equação 2.11 no Lagrangeano do problema primal, Equação 2.9, pode-se obter o problema dual da SVM, Equação (2.13):

$$\begin{aligned} L(\mathbf{w}, b, \alpha) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n a_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^n a_i y_i b + \sum_{i=1}^n a_i \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n a_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^n a_i y_i b + \sum_{i=1}^n a_i \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n a_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^n a_i y_i b + \sum_{i=1}^n a_i \end{aligned} \quad (2.12)$$

$$\begin{aligned}
\text{□ Maximizar: } L(a) &= -\frac{1}{2} \sum_{i=1}^n a_i a_i y_i y_i \mathbf{x}_i^T \mathbf{x}_i + \sum_{i=1}^n a_i \\
\text{□ Sujeito a: } &\sum_{i=1}^N a_i y_i = 0 \\
&a_i \geq 0 \text{ para } i = 1, \dots, n
\end{aligned} \tag{2.13}$$

Vale notar que a função de decisão pode ser descrita pela adição do resultado obtido na Equação (2.11) por:

$$f(\mathbf{x}) = \text{sin}(\sum_{i=1}^N a_i y_i \mathbf{x}_i^T \mathbf{x} + b). \tag{2.14}$$

2.2.2 SVM - Margem flexível

É sabido que muitos dos problemas reais de classificação não são linearmente separáveis, portanto é necessário uma formulação para SVMs que leve em consideração esse tipo de problema. Essa nova formulação consiste na adição de variáveis de folga com a finalidade de relaxar as restrições do problema de otimização, resultando em

$$\begin{aligned}
y_i[\mathbf{w}_o^T \mathbf{x}_i + b_o] &\geq 1 - \varepsilon_i \\
\varepsilon_i &\geq 0 \text{ e } i, \dots, n.
\end{aligned} \tag{2.15}$$

Um importante ponto a ser levantado é o significado dos valores das variáveis de folga. Um vetor $\mathbf{x}_i$ corretamente classificado e longe da margem possui um valor de ε_i igual a zero. Um vetor localizado entre a margem e o hiperplano de decisão e corretamente classificado possui um valor de ε_i no intervalo de $(0, 1]$. Finalmente o vetor localizado na região da classe oposta possui um valor de $\varepsilon_i > 1$.

Ao considerar o uso de variáveis de folga, pode-se representar o problema primal por

$$\begin{aligned}
\text{□ Minimizar: } J(\mathbf{w}, \boldsymbol{\varepsilon}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_i \varepsilon_i \\
\text{□ Sujeito a: } &y_i[\mathbf{w}_o^T(\mathbf{x}_i) + b_o] - 1 + \varepsilon_i \geq 0 \\
&\varepsilon_i \geq 0, i = 1, \dots, n
\end{aligned} \tag{2.16}$$

em que γ é um fator de regularização que implica diretamente na complexidade do modelo e no controle do erro empírico. Um valor grande de γ aumenta o erro de treinamento, enquanto o valor muito pequeno leva a um comportamento similar à SVM com margem rígida. Após utilizar o Lagrangeano obtem-se:

$$L(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i y_i [\mathbf{w}_i^T \mathbf{x}_i + b] - 1 + \varepsilon_i - \sum_{i=1}^n \beta_i \varepsilon_i \tag{2.17}$$

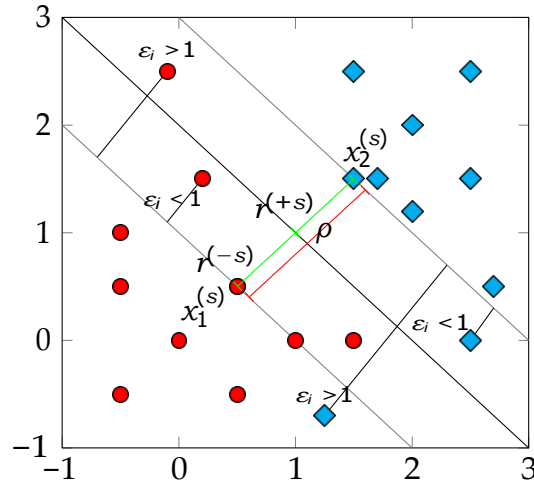


Figura 3 – Interpretação da SVM com margens flexíveis. Perceba que ao permitir que alguns padrões sejam classificados de forma errada, como por exemplo o ponto azul mais inferior, é possível escolher um hiperplano que classifique corretamente a grande maioria dos pontos.

A solução $(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ consiste em, de maneira similar às SVM de margem rígida, encontrar o ponto de sela da função $L(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. em que se busca

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \min_{\mathbf{w}, b, \boldsymbol{\varepsilon}} L(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta}). \quad (2.18)$$

Assim, para a resolução do problema de otimização devem ser consideradas as seguintes condições de optimalidade, a saber:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i; \quad (2.19)$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = 0 \rightarrow \sum_{i=1}^N \alpha_i y_i = 0; \quad (2.20)$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{\varepsilon}} = 0 \rightarrow \gamma = \alpha_i + \beta_i. \quad (2.21)$$

Após algumas manipulações na função $L(\mathbf{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ é possível substituir os valores obtidos nas Equações (2.19), (2.20) e (2.21) para chegar a formulação do problema dual

$$\begin{aligned} \square \text{Maximizar} \quad & L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^N \alpha_i \\ \square \text{Sujeito a} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq \gamma \quad \text{para } i = 1, \dots, n \end{aligned} \quad (2.22)$$

O problema dual obtido pode ser resolvido através de técnicas de programação quadrática.

Como é sabido, diversas rotinas para a solução deste problema de otimização estão presentes em pacotes de *softwares* como Matlab, Scilab e Octave. Outra opção geralmente utilizada é a solução do problema pelo algoritmo SMO (*Sequential Minimal Optimization*) desenvolvido por Platt(1999). O algoritmo SMO resolve o problema iterativamente pela otimização de sub-problemas de programação quadrática. Essa técnica é mais utilizada atualmente e está presente, por exemplo no pacote LIBSVM¹.

Vale ressaltar que a solução para a SVM de margem flexível difere apenas pela a adição da restrição $a_i \leq \gamma$ ao problema de otimização.

2.3 Máquinas de Vetores Suporte para Regressão

A mesma ideia, aplicada à problemas de classificação para SVM, pode ser aplicada também para estimar funções lineares e não lineares, sendo necessário apenas algumas modificações na formulação anteriormente apresentada. A primeira delas diz respeito ao hiperplano de separação entre as classes, no qual em problemas de regressão se traduz na função $f(\mathbf{x})$, estimada pelo modelo. A segunda são as margens de classificação que devem ser maximizadas pela SVM, essas são semelhantes ao que é chamado de tubo, definido pela SVR. Para a formulação matemática da SVR considere o conjunto de treinamento $\{\mathbf{x}_i, y_i\}_{i=1}^N$ utilizado para estimar a função $f(\mathbf{x})$ apresentada a seguir.

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \quad (2.23)$$

A minimização do erro empírico, de acordo com Vapnik (VAPNIK, 2000) segue a seguinte função de custo:

$$R_{emp} = \frac{1}{N} \sum_{i=1}^N |y_i - \mathbf{w}^T \mathbf{x}_i - b|_s \quad (2.24)$$

em que $|y_i - f(\mathbf{x})|_s$ representa uma função de custo s (s -intensive loss function).

$$|y - f(\mathbf{x})|_s = \begin{cases} 0, & \text{se } |y - f(\mathbf{x})| \leq s \\ |y - f(\mathbf{x})| - s, & \text{se } |y - f(\mathbf{x})| > s \end{cases} \quad (2.25)$$

Pela análise da Equação (2.25) pode-se perceber que a SVR têm como objetivo estimar uma função $f(\mathbf{x})$ com no máximo $|s|$ de erro em relação às saídas desejadas y_i do conjunto de treinamento. Em outras palavras, um erro menor que $|s|$ é desconsiderado e um erro maior que $|s|$ é computado de forma linear. Nas Figuras 4(a) e 4(b) são apresentados o significado geométrico da função apresentada na Equação (2.25). Na Figura 4(a), a reta estimada, linha contínua, possui um erro máximo de $\pm s$ em relação a todos os padrões;

¹ Uma das bibliotecas mais comuns para treinar e executar a SVM atualmente.

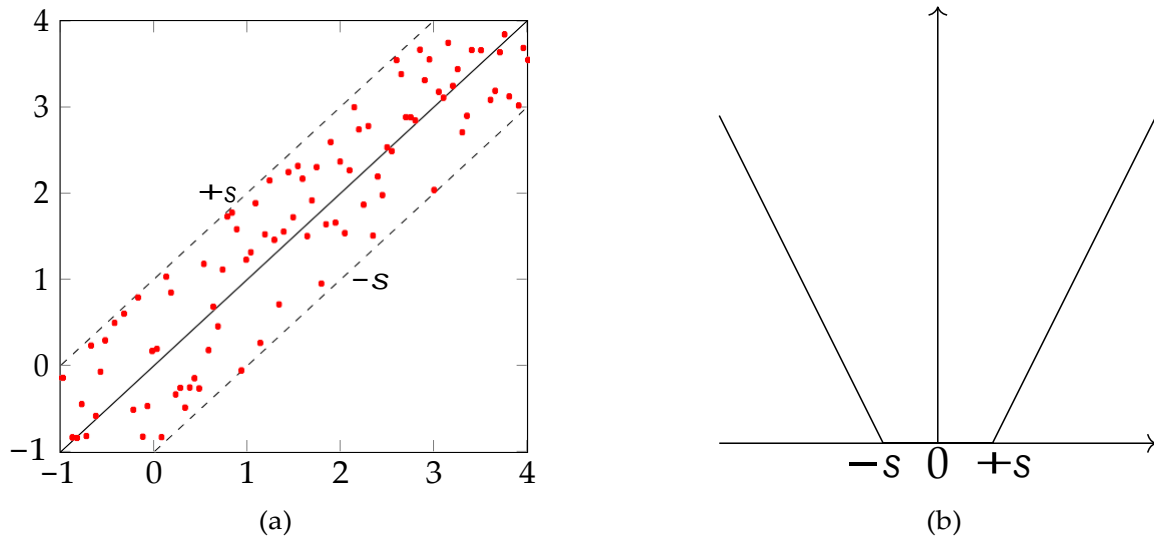


Figura 4 – A linha contínua na Figura a) é apresentada, a função estimada pelo s -SVR. A linha pontilhada define as margens correspondentes a um erro de $\pm s$. Na Figura b) é apresentada a forma da função proposta por Vapnik.

enquanto a reta pontilhada representa a margem que define o tubo maximizado pela SVR. Na Figura 4(b) a Equação (2.25) é apresentada graficamente indicando que qualquer amostra $\mathbf{x}_i$ com um erro menor que $|s|$ é considerado como igual a zero. Uma vez que o principal parâmetro de configuração da SVR é o s , essa versão é chamada de s -SVR. Uma outra versão, proposta por Schölkopf *et al.* (2000), tira o foco do parâmetro s e o transfere para um novo parâmetro que controla o número de vetores suporte.

De maneira similar à apresentada para SVM de margem rígida na seção anterior, pode-se maximizar a largura do tubo pela minimização de $J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$. Uma vez que ainda se considera as restrições, apresentadas na Equação (2.25), pode-se chegar a seguinte formulação primal do problema de otimização para um problema de regressão:

$$\begin{aligned}
 &\square \text{ Minimizar} && J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \\
 &\text{Sujeito a} && y_i - \mathbf{w}^T \mathbf{x}_i \leq s, i = 1, \dots, N \\
 &\square && \mathbf{w}^T \mathbf{x}_i + b - y_i \leq s, i = 1, \dots, N
 \end{aligned} \tag{2.26}$$

Um ponto importante a ser ressaltado diz respeito a escolha do valor de s , um s pequeno pode tornar o problema sem solução enquanto um s grande pode resultar em infinitas soluções assemelhando-se às redes neurais convencionais nas quais não há preocupação com a margem de separação. Na Figura 5 são apresentadas situações para diferentes valores de s .

Na Figura 6(a), a reta encontrada não está bem ajustada aos pontos, apesar de nenhuma restrição ter sido violada. Essa condição é interessante, pois a partir desse ponto é possível derivar o problema de otimização para s -SVR com margem flexível. veja que

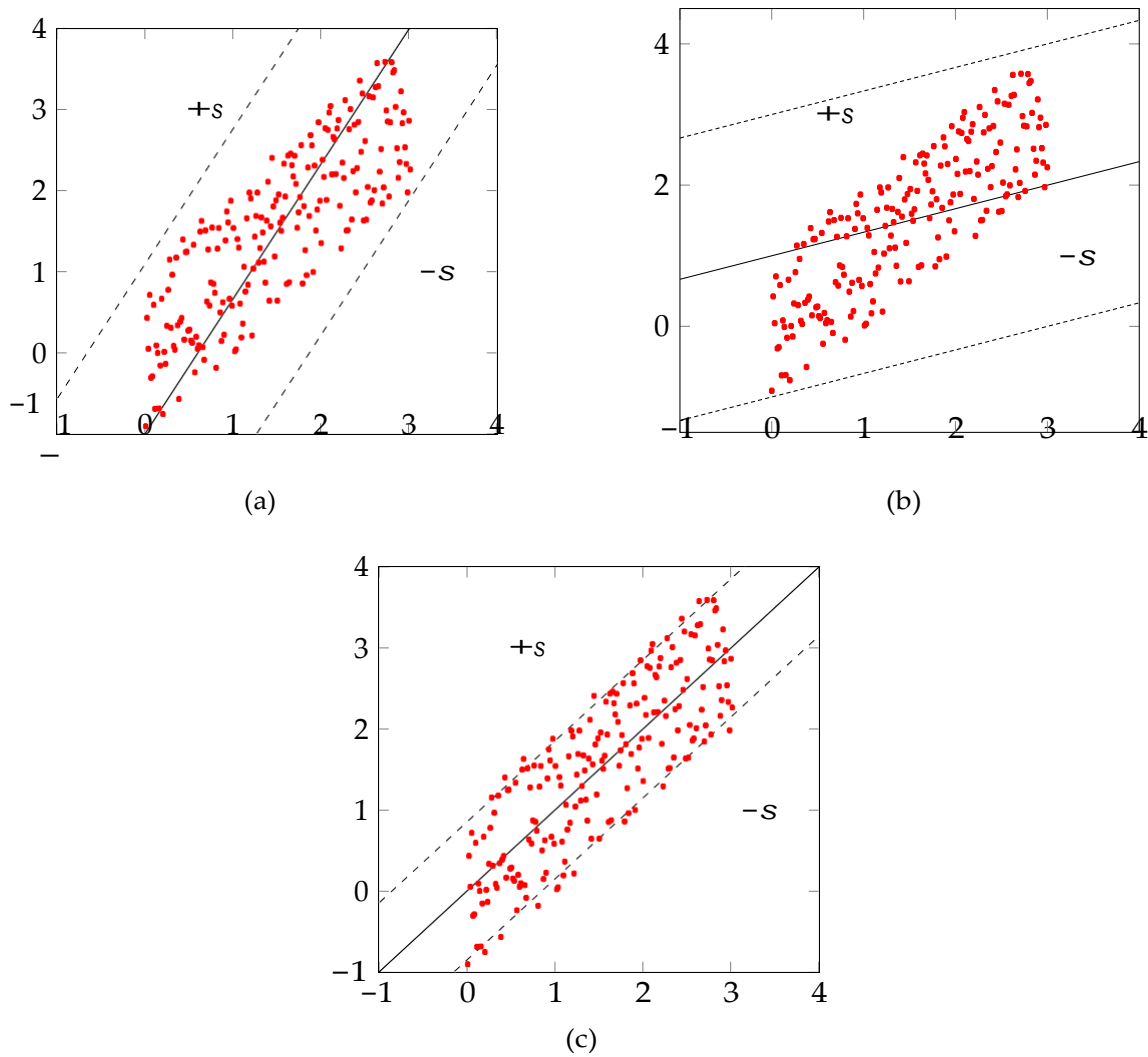


Figura 5 – Diversos valores de s para um mesmo conjunto de dados. Na Figura a) e b) percebe-se que o valor de s é maior que o necessário para criar um tubo adequado ao redor dos dados, permitindo que tal tubo se posicione de várias formas sem acompanhar a dinâmica dos dados. Na Figura c) um valor pequeno de s torna o problema sem solução uma vez que algumas das restrições serão sempre violadas.

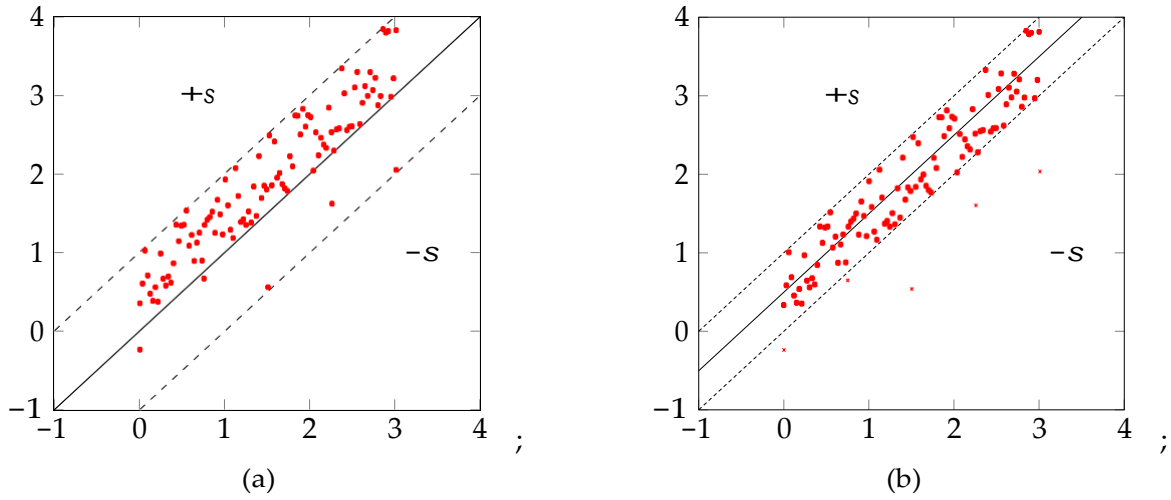


Figura 6 – Resultado obtido com a flexibilização da margem do tubo. a) A reta em preto, não pontilhada, não está bem ajustada aos pontos mesmo assim não viola as restrições apresentadas na Equação 2.27. b) Ao ignorar alguns dos pontos obtém-se uma função melhor ajustada.

se for possível ignorar alguns dos padrões, principalmente os que estão abaixo da reta preta, não pontilhada, será possível ajustar um novo hiperplano de forma mais justa. Essa situação é ilustrada na Figura 6(b). A fim de permitir um melhor ajuste faz-se uso de variáveis de folga, ξ_i e ξ_i^* . Essas variáveis de folga são adicionadas às margens a fim de permitir que alguns padrões fiquem fora do tubo, desta forma, tem-se o seguinte problema primal de otimização.

$$\begin{aligned}
 &\square \text{Minimizar} \quad J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_{i=1}^N (\xi_i + \xi_i^*) \\
 &\square \text{Sujeito a} \quad y_i - \mathbf{w}^T \mathbf{x}_i - b \leq s + \xi_i, \quad i = 1, \dots, N \\
 &\quad \quad \quad \mathbf{w}^T \mathbf{x}_i + b - y_i \leq s + \xi_i^*, \quad i = 1, \dots, N \\
 &\square \quad \quad \quad \xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, N
 \end{aligned} \tag{2.27}$$

O segundo termo a ser minimizado corresponde ao erro total inserido quando adicionam-se as variáveis de folga. Isso pode ser identificado ao se verificar que ξ_i e ξ_i^* não podem, simultaneamente, ter seus valores iguais a zero. De maneira semelhante ao exposto para a otimização das SVM, pode-se expressar o problema primal por

$$\max_{\alpha, \alpha^*, \eta, \eta^*} \min_{\mathbf{w}, b, \xi, \xi^*} L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*) \tag{2.28}$$

em que

$$\begin{aligned}
 L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*) = & \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_{i=1}^N (\xi_i + \xi_i^*) \\
 & + \sum_{i=1}^N a_i (s + \xi_i - y_i + \mathbf{w}^T \mathbf{x}_i + b) \\
 & + \sum_{i=1}^N \alpha_i^* (\xi_i^* - \xi_i) + \sum_{i=1}^N \eta_i \xi_i - \sum_{i=1}^N \eta_i^* \xi_i^*
 \end{aligned} \quad (2.29)$$

em que a_i , α_i^* , η_i e η_i^* são os multiplicadores de Lagrange.

Para solucionar o problema supracitado, a função $L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*)$ deve ser otimizada em relação a $\mathbf{w}, b, \xi, \xi^*$, para tal, são obtidas as expressões para as condições de otimalidade da função apresentada na Equação (2.29), ou seja,

$$\frac{\partial L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*)}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N (a_i - \alpha_i^*) \mathbf{x}_i \quad (2.30)$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*)}{\partial b} = 0 \Rightarrow \sum_{i=1}^N (a_i^* - a_i) = 0 \quad (2.31)$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*)}{\partial \xi_i} = 0 \Rightarrow \eta_i = \gamma - a_i \quad (2.32)$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*)}{\partial \xi_i^*} = 0 \Rightarrow \eta_i^* = \gamma - \alpha_i^* \quad (2.33)$$

Pela substituição dos termos encontrados nas Equações 2.30, 2.31, 2.32 e 2.33 na Equação 2.29, chega-se a seguinte definição do problema dual de otimização:

$$\begin{aligned}
 & \square \quad \text{Maximizar:} \quad L(\alpha, \alpha^*) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (a_i - \alpha_i^*)(a_j - \alpha_j^*) \mathbf{x}_i^T \mathbf{x}_j \\
 & \quad \quad \quad - \sum_{i=1}^N (a_i + \alpha_i^*) y_i (a_i - \alpha_i^*) \\
 & \square \quad \text{Sujeito a:} \quad \sum_{i=1}^N (a_i - \alpha_i^*) = 0 \\
 & \quad \quad \quad 0 \leq a_i \leq \gamma \\
 & \quad \quad \quad 0 \leq \alpha_i^* \leq \gamma
 \end{aligned} \quad (2.34)$$

Similarmente, pela substituição de $\mathbf{w} = \sum_{i=1}^N (a_i - \alpha_i^*) \mathbf{x}_i$, na Equação (2.23), é obtida a função para estimação da saída para um determinado padrão $\mathbf{x}$ mostrada a seguir.

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N \alpha_i \phi(\mathbf{x}_i) \quad (2.35)$$

2.4 Truque de Kernel

Apesar das versões com margens flexíveis das máquinas de vetores suporte serem capazes de resolver problemas não lineares, o modelo final obtido por elas é caracterizado por um hiperplano, dessa forma sua capacidade de representar os padrões de treinamento é limitada.

Isso pode ser verificado ao se observar as Equações (2.14) e (2.35), as quais representam uma combinação linear dos vetores suporte. Através de um mapeamento não linear, substituindo $\mathbf{x}_i$ por $\phi(\mathbf{x}_i)$ pode-se tratar problemas de classificação ou regressão mais complexos. De acordo com o teorema de Mercer (1909), uma função simétrica e semi positiva $K(\mathbf{x}_i, \mathbf{x}_j)$ pode ser expressa em termos do seguinte produto interno.

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\phi(\mathbf{x}_i) \phi(\mathbf{x}_j)) \quad (2.36)$$

Isso é interessante pois pode-se aplicar o chamado, truque de *kernel*, de maneira direta ao problema da SVM, e assim, tratar de problemas mais complexos. O truque de *kernel* consiste em realizar a substituição do produto $\mathbf{x}^T \mathbf{x}_i$ por $K(\mathbf{x}, \mathbf{x}_i)$, presentes nas equações (2.14) e (2.35), as quais estimam o hiperplano de decisão e a função de regressão, respectivamente. Além disso é necessária a substituição também nas equações (2.22) e (2.34) do problema dual das SVCs e SVRs respectivamente. Os *kernels* mais comumente utilizados são o linear, polinomial e gaussiano (RBF). Estes *kernels* e o *kernel* MLP são apresentados na Tabela 1,

Tabela 1 – Funções de *kernel* tipicamente utilizadas em máquinas de vetores suporte.

Tipo de Kernel	Função $K(\mathbf{x}, \mathbf{x}_i)$	Descrição
Linear	$K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}^T \mathbf{x}_i$	A função é estimada utilizando apenas os próprios dados.
Polinomial	$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i)^p$	O parâmetro p especifica o grau do polinômio.
RBF	$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\ \mathbf{x} - \mathbf{x}_i\ _2^2 / \sigma^2)$	σ especifica a largura da função RBF.
MLP	$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1)$	β_0 especifica a inclinação e β_1 especifica o deslocamento da tangente hiperbólica.

2.5 Máquinas de Vetores Suporte de Mínimos Quadrados para Classificação

Nesta sessão será discutido as máquinas de vetores suporte de mínimos quadrados (*Least Squares Support Vector Machines*, LS-SVM). A LSSVM é uma alternativa para a formulação padrão da SVM, basicamente transforma o problema primal, baseado em programação quadrática, em um formato que pode ser resolvido pelos mínimos quadrados (SUYKENS; VANDEWALLE, 1999).

Considere o problema primal da SVM com margem flexível que é exposto abaixo:

$$\begin{aligned} \square \text{Minimizar: } J(\mathbf{w}, \boldsymbol{\varepsilon}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_{i=1}^n \varepsilon_i \\ \square \text{Sujeito a: } y_i [\mathbf{w}_o^T(\mathbf{x}_i) + b_o] - 1 + \varepsilon_i &\geq 0 \\ \square \varepsilon_i &\geq 0, i = 1, \dots, n \end{aligned} \quad (2.37)$$

A LSSVM reformula o problema primal de otimização, apresentado anteriormente, realizando basicamente 3 modificações em sua formulação padrão. A primeira delas é a substituição das restrições que antes eram de desigualdade e passaram a ser de igualdade. Essa alteração obriga a remoção da última restrição, uma vez que para ser possível resolver o problema as variáveis de folga devem assumir qualquer valor. Por isso o que antes eram variáveis de folga na formulação da SVM passam a ser chamadas de variáveis de erro na formulação da LSSVM. A última modificação é que no termo de regularização o valor de γ pondera o erro quadrático, resultando em:

$$\begin{aligned} \square \text{Minimizar: } J(\mathbf{w}, \mathbf{e}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{i=1}^n e_i^2 \\ \square \text{Sujeito a: } y_i [\mathbf{w}_o^T(\mathbf{x}_i) + b_o] &= 1 - e_i \end{aligned} \quad (2.38)$$

De maneira similar a SVM pode-se utilizar o Lagrangeano para obter o problema dual da LSSVM. Assim tem-se:

$$L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{i=1}^n e_i^2 - \sum_{i=1}^n \alpha_i (y_i [\mathbf{w}^T \mathbf{x}_i + b] - 1 + e_i) \quad (2.39)$$

em que α_i são os multiplicadores de Lagrange. Uma vez que as restrições são de igualdade, os multiplicadores de Lagrange podem assumir valores positivos ou negativos. Dessa forma, as condições de optimalidade, que correspondem as derivadas parciais em relação as quatro variáveis, $\mathbf{w}$, b , $\mathbf{e}$, $\boldsymbol{\alpha}$, podem ser calculadas.

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \rightarrow \sum_{i=1}^N a_i y_i \mathbf{x}_i = 0 \quad (2.40)$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial b} = 0 \rightarrow \sum_{i=1}^N a_i y_i = 0 \quad (2.41)$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial \mathbf{e}} = 0 \rightarrow y e_i = \alpha_i \quad (2.42)$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial \alpha} = 0 \rightarrow y [\mathbf{w}^T \mathbf{x}_i + b] - 1 + e_i = 0 \quad (2.43)$$

Com base nessas condições, pode-se definir o seguinte sistema linear

$$\mathbf{D}\mathbf{z} = \mathbf{t}, \quad (2.44)$$

em que

$$\mathbf{D} = \begin{array}{c|ccc} & \mathbf{y}^T & & \\ \hline 0 & \mathbf{\Omega} + \mathbf{y}^{-1}\mathbf{I} & \mathbf{b} & 0 \\ \hline \mathbf{y} & \mathbf{\alpha} & \mathbf{e} & \mathbf{1} \end{array} \quad \mathbf{z} = \begin{bmatrix} \mathbf{w} \\ b \\ \mathbf{e} \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} 0 \\ \mathbf{b} \\ \mathbf{1} \end{bmatrix} \quad (2.45)$$

Na matriz $\mathbf{D}$, $\mathbf{I}$ representa a matriz identidade de tamanho N , enquanto o símbolo $\mathbf{1}$ representa o vetor N -dimensional com todos os valores iguais a 1 e, finalmente:

$$\Omega_{i,j} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j \mid i, j = 1, \dots, N \quad (2.46)$$

Note que o produto da primeira linha de $\mathbf{D}$ por $\mathbf{z}$ representa a segunda condição de optimalidade, Equação (2.41) e o produto da segunda, representa a última condição, Equação (2.43).

A solução $\mathbf{z}$ pode ser obtida pelo métodos dos mínimos quadrados:

$$\mathbf{z} = \mathbf{D}^{-1}\mathbf{t}. \quad (2.47)$$

Após cálculo dos valores dos multiplicadores de Lagrange, do intercepto e da inserção da condição de otimalidade apresentada na Equação (2.40) na função discriminante, chega-se a seguinte expressão.

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N a_i y_i \mathbf{x}^T \mathbf{x}_i + b \right). \quad (2.48)$$

Da mesma forma que dito na Sessão 2.4, o truque de *kernel* pode ser aplicado na função discriminante, Equação 2.48 e no cômputo de $\Omega_{i,j}$, Equação 2.46. Para isso deve-se trocar o produto interno $\mathbf{x}^T \mathbf{x}_i$ por alguma função de *kernel* $K(\mathbf{x}, \mathbf{x}_i)$.

2.6 Máquinas de Vetores Suporte de Mínimos Quadrados para Regressão

De maneira muito semelhante às LSSVMs pode-se obter a formulação para as LSSVRs. Considerando o problema primal da SVR tem-se:

$$\begin{aligned}
 &\square \text{Minimizar: } J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_{i=1}^N (\xi_i + \xi_i^*) \\
 &\square \\
 &\square \text{Sujeito a: } y_i - \mathbf{w}^T \mathbf{x}_i - b \leq s + \xi_i, i = 1, \dots, N \\
 &\quad \mathbf{w}^T \mathbf{x}_i + b - y_i \leq s + \xi_i^*, i = 1, \dots, N \\
 &\square \quad \xi_i, \xi_i^* \geq 0, i = 1, \dots, N
 \end{aligned} \tag{2.49}$$

A LSSVR reformula-o da seguinte maneira:

$$\begin{aligned}
 &\square \text{Minimizar: } J(\mathbf{w}, \mathbf{e}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{i=1}^N e_i^2 \\
 &\square \text{Sujeito a: } y = \mathbf{w}_o^T(\mathbf{x}) + b + e_i
 \end{aligned} \tag{2.50}$$

De maneira idêntica à LSSVM, a variável e_i pode assumir valores negativos e positivos com isso pode-se trocar todas as restrições do problema por apenas uma. Uma vez que, sujeito a dimensionalidade de $\mathbf{w}$ pode ser mais conveniente resolver o problema na forma dual, ao aplicar o método de Lagrange à Equação (2.50) obtém-se:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_{i=1}^N e_i^2 - \sum_{i=1}^N \alpha_i (\mathbf{x}_i^T \mathbf{w} + b - y_i + e_i). \tag{2.51}$$

em que $\{\alpha_i\}_{i=1}^N$ são os multiplicadores de Lagrange. A fim de resolver o problema, a função $L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})$ deve ser otimizada em relação a $\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha}$. Assim, obtem-se as seguintes condições de optimalidade.

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i \mathbf{x}_i, \tag{2.52}$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial b} = 0 \Rightarrow \sum_{i=1}^N \alpha_i = 0, \tag{2.53}$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial \alpha_i} = 0 \Rightarrow (\mathbf{x}_i^T \mathbf{w} + b) - y_i + e_i = 0, \tag{2.54}$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha})}{\partial e_i} = 0 \Rightarrow \alpha_i = \gamma e_i. \tag{2.55}$$

Com base nessas condições, pode-se definir o seguinte sistema linear

$$\mathbf{Dz} = \mathbf{t}, \tag{2.56}$$

em que

$$\mathbf{D} = \begin{array}{c|c} 0 & \mathbf{1}^T \\ \hline \mathbf{1} & \Phi + \gamma^{-1} \mathbf{I} \end{array}, \mathbf{z} = \begin{array}{c} b \\ \alpha \end{array}, \mathbf{t} = \begin{array}{c} 0 \\ \mathbf{y} \end{array}, \quad (2.57)$$

tal que $\mathbf{I}$ é uma matriz identidade de tamanho N , o símbolo $\mathbf{1}$ representa um vetor de 1s N -dimensional e $\Phi_{i,j} = \mathbf{x}_i^T \mathbf{x}_j$.

A solução $\mathbf{z}$ pode ser obtida através do método dos mínimos quadrados

$$\mathbf{z} = \mathbf{D}^{-1} \mathbf{t}. \quad (2.58)$$

Após a obtenção dos valores dos multiplicadores de Lagrange, do intercepto e pela adição da condição de optimalidade apresentada na Equação 2.52 na função $f(x)$ da SVR, Equação 2.2, obtem-se a nova função $f(x)$:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \mathbf{x}^T \mathbf{x}_i + b. \quad (2.59)$$

Da mesma forma do exposto para o problema de classificação, pode-se aplicar o truque de *kernel* a função discriminante, Equação 2.48 e no cômputo de $\Omega_{i,j}$. Para isto basta que o produto interno $\mathbf{x}^T \mathbf{x}_i$ seja substituído por alguma função de *kernel* $K(\mathbf{x}, \mathbf{x}_i)$.

2.7 Máquinas de Vetores Suporte Esparsas

Na seção anterior, são apresentadas as principais características e formas de treinamento disponíveis para as máquinas de vetores suporte. Nesta seção, por sua vez, são apresentadas soluções ao problema decorrente da utilização dos mínimos quadrados na solução do problema de otimização das LSSVMs. Em contrapartida à simplificação imposta ao problema, as LSSVMs não conseguem prover uma solução esparsa, isto é, uma vez que os multiplicadores de Lagrange, normalmente, não possuem valores iguais a zero, os vetores suporte associados à cada um dos padrões devem ser armazenados a saída desejada. Em seguida, são apresentados métodos que visam reduzir o número de vetores suporte.

2.7.1 P-LSSVM

O Pruning LSSVM(P-LSSVM), foi proposto por Suykens, Lukas e Vandewalle (2000a). Neste método os vetores suporte são removidos de acordo com os valores dos multiplicadores de Lagrange. O processo é realizado de forma recursiva, de tal maneira que os vetores são eliminados iterativa e gradualmente o critério de parada utilizado é a perda de desempenho frente a um conjunto de validação. O processo de treinamento pode ser descrito como segue:

1. O sistema linear apresentado na Equação (2.45) para classificação ou na Equação (2.57) para regressão é resolvido²
2. O erro é calculado utilizando o conjunto de validação. É importante ressaltar que para problemas de regressão o erro quadrático médio é utilizado.
3. O parametro $\tau \in [0, 1]$ deve ser definido para representar o percentual de vetores suporte que serão removidos durante cada etapa de remoção. Neste ponto Suykens, Lukas e Vandewalle (2000b) apenas sugerem uma remoção baseada em um percentual, alternativamente pode ser removida uma quantidade constante de vetores ou variável seguindo alguma regra definida de forma empírica.
4. O parametro ξ define a tolerância ao erro em relação ao mínimo erro obtido durante o treinamento.
5. Os valores absolutos dos multiplicadores de Lagrange obtidos do vetor $\mathbf{z}$ são ordenados. Note que não deve ser considerado o primeiro elemento do vetor $\mathbf{z}$, pois refere-se ao viés ou intercepto.
6. A porção τ com os menores valores absolutos dos multiplicadores de Lagrange e seus vetores suporte são então removidos. Na prática, a remoção de um vetor suporte equivale a remoção das colunas associadas a esses vetores em $\mathbf{D}$. Veja que para a remoção do i -ésimo vetor suporte deve ser removido a coluna $(i+1)$.
7. Seja a matriz $\mathbf{D}_*$ construída com os vetores suporte não removidos, o novo sistema linear, representado por $\mathbf{D}_* \mathbf{z}_* = \mathbf{t}_*$, é solucionado em $\mathbf{z}_*$ como sendo $\mathbf{z}_* = \mathbf{D}_*^\dagger \mathbf{t}_*$, onde $\mathbf{D}_*^\dagger$ é a pseudo-inversa de Moore-Penrose (BEN-ISRAEL, 2003) e pode ser computada como:
$$\mathbf{D}_*^\dagger = (\mathbf{D}_*^T \mathbf{D}_*)^{-1} \mathbf{D}_*^T. \quad (2.60)$$
8. O vetor $\boldsymbol{\alpha}$ e o intercepto b são obtidos a partir da solução $\mathbf{z}_*$.
9. Em posse dos multiplicadores de Lagrange $\boldsymbol{\alpha}$ e o intercepto b é possível estimar o erro do conjunto de validação. Isso pode ser realizado, em problemas de classificação, ao substituir estes valores na função discriminante, Equação (2.48). De maneira similar, desta vez para o problema de regressão, a substituição de $\boldsymbol{\alpha}$ e b pode ser feita na Equação (2.59).
10. O erro obtido com o conjunto de validação é avaliado. Se o erro é maior que a tolerância ξ , então o critério de convergência é alcançado e, portanto, o processo de remoção é finalizado. Caso contrário, voltar ao passo 5 e usar $\mathbf{z}_*$ em vez de $\mathbf{z}$.

² A matriz $\mathbf{D}$ é obtida com base em todos os padrões de treinamento

Como a P-LSSVM é um método que impõe esparsidade aos modelos e possui dois parâmetros de controle, τ e ε , em que o primeiro parâmetro controla a quantidade de vetores que devem ser removidos por iteração, um valor atribuído a este tem um forte impacto no tempo de treinamento. Um valor τ pequeno realiza a poda em porções menores de vetores suporte a cada iteração, levando ao cômputo de várias soluções de sistema lineares com matrizes pouco reduzidas. Enquanto um valor τ grande reduz o número de iterações ao passo que as matrizes são reduzidas mais rapidamente a cada iteração. Essa segunda alternativa, porém, pode resultar em um modelo não tão esparsa em relação à primeira, uma vez que essa é realizada de forma mais refinada.

Além disso, por ser uma técnica de poda e seu treinamento ser realizado a partir de um conjunto inicial contendo todos os vetores suporte, os passos iniciais do treinamento são computacionalmente intensos, e para um conjunto de dados suficientemente grande, a P-LSSVM se torna computacionalmente inviável.

2.7.2 IP-LSSVM

A IP-LSSVM (CARVALHO; BRAGA, 2009) utiliza os valores associados aos multiplicadores de Lagrange para selecionar um conjunto de vetores suporte durante o treinamento. Após o cálculo dos multiplicadores de Lagrange estes serão ordenados e os vetores suportes associados aos multiplicadores mais próximos de zero são removidos.

O método parte da idéia de que um padrão que possua valor de multiplicador $a_i = 0$ tem uma maior probabilidade de se tornar um vetor suporte, pois, de acordo com Carvalho, tal padrão encontra-se mais próximo da margem, foi proposto para a IP-LSSVM os seguintes critérios de seleção de vetores suporte. Veja que esta idéia tem relação com o conceito de vetores suporte nas máquinas de vetores suporte tradicionais.

- Se $a_i = 0$, $\mathbf{x}_i$ será um vetor suporte caso este esteja localizado próximo ao hiperplano de separação das classes ou localizado na região da outra classe, em que $a_i = 0$ pode ser definido como os maiores multiplicadores obtidos. Em problemas de regressão $a_i = 0$ significa que o vetor $\mathbf{x}_i$ está longe e acima da função estimada. Esses vetores correspondem aos $a_i = 0$ utilizados no QP-SVM.
- Se $a_i \geq 0$, $\mathbf{x}_i$ será removido no caso de ser corretamente classificado, indicando que está longe do hiperplano de separação das classes, porém próximo as margens. Quando o vetor está sobre a margem $a_i = 0$. Em problemas de regressão isso significa que os vetores estão entre as margens positivas e negativas. Quando o vetor está sobre a função estimada $a_i = 0$.
- Se $a_i < 0$ ou $a_i = 0$, $\mathbf{x}_i$ será removido. Esses valores de a_i indicam que $\mathbf{x}_i$ foi corretamente classificado e é equivalente ao $a_i = 0$ associado ao QP-SVM. Em

problemas de regressão $\alpha_i = 0$ significa que o vetor $\mathbf{x}_i$ está longe e abaixo da função estimada.

O processo de treinamento dos IP-LSSVMs é apresentado a seguir.

1. O sistema linear de equações apresentado na Equação (2.45) para classificação ou na Equação (2.57) para regressão é resolvido.
2. O parametro $\tau \in [0, 1]$ deve ser definido para representar o percentual de vetores suporte que serão removidos durante cada etapa de remoção.
3. Os multiplicadores de Lagrange são ordenados e a fração τ , contendo os menores α_i e seus vetores suporte associados são removidos. Em problemas de regressão é necessário avaliar $|\alpha_i|$, caso contrário serão removidos apenas os vetores suporte que estão abaixo da curva estimada.
4. A fração $1 - \tau$ restante é utilizada para calcular os novos multiplicadores de Lagrange.
5. A matriz não quadrada $\mathbf{D}_*$ é contruída com os vetores suporte não removidos e o novo sistema linear de equações, representado por $\mathbf{D}_* \mathbf{z}_* = \mathbf{t}_*$, é solucionado $\mathbf{z}_* = \mathbf{D}_*^\dagger \mathbf{t}_*$, em que $\mathbf{D}_*^\dagger$ é a pseudo-inversa de Moore-Penrose e pode ser computada como

$$\mathbf{D}_*^\dagger = (\mathbf{D}_*^T \mathbf{D}_*)^{-1} \mathbf{D}_*^T. \quad (2.61)$$
6. O vetor $\boldsymbol{\alpha}$ e o intercepto b são obtidos a partir da solução $\mathbf{z}_*$.

As IP-LSSVMs permitem identificar os vetores suporte próximos à margem de separação sem perder a simplicidade da LSSVM. Além disso uma LSSVM é um método de baixo custo computacional, uma vez que o processo de treinamento se resume a duas inversões de matrizes, mostrando-se assim uma solução adequada para impor esparsidade à modelos.

2.8 Conclusão

Neste capítulo foi apresentada a formulação das Máquinas de Vetores Suporte, como ocorre a maximização da margem de separação entre as classes em problemas de classificação e a maximização do tubo *s-intensive* em problemas de regressão. Foram discutidas as versões lineares e não lineares para classificadores e regressores, além de mostrar como utilizar o truque de *kernel* para possibilitar melhores resultados no tratamento de problemas não lineares. Foi discutido também a formulação das Máquinas de Vetores Suporte de Mínimos Quadrados para classificação e regressão, mostrando a origem do problema de esparsidade do modelo. Finalmente foram apresentadas duas técnicas para

obtenção de modelos esparsos na LSSVM, a P-LSSVM, uma técnica iterativa e de alto custo computacional e a IP-LSSVM, outra técnica para obtenção de um modelo menor com um custo computacional bem inferior.

3 Redução de Modelos Lineares

3.1 Introdução

Neste capítulo são apresentadas técnicas para redução de modelos, quando o mesmo é obtido pela resolução de um sistema linear $\mathbf{D}\mathbf{z} = \mathbf{t}$. Esta redução tem como finalidade encontrar um subconjunto de variáveis, pertencentes ao conjunto de todas as variáveis (colunas) da matriz $\mathbf{D}$, que melhor expliquem a variável dependente $\mathbf{t}$. Portanto, uma variável é formada pelos valores de uma componente específica contidas nas diversas instâncias (padrões) que compõe o conjunto de treinamento. Assim uma linha da matriz $\mathbf{D}$ representa uma instância (padrão). Na Equação (3.1) Considerando $\mathbf{t}$ um vetor unidimensional, z_i é um valor escalar associado a i -ésima dimensão de $\mathbf{D}$, dessa forma, denota-se $\mathbf{d}_i$ como sendo um vetor coluna associado a i -ésima coluna da matriz $\mathbf{D}$.

$$\begin{bmatrix} \mathbf{d}_1 & \dots & \mathbf{d}_i & \dots & \mathbf{d}_p \end{bmatrix}_{N \times p} \begin{bmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_p \end{bmatrix}_{p \times 1} = \begin{bmatrix} t_1 \\ \vdots \\ t_j \\ \vdots \\ t_N \end{bmatrix}_{N \times 1} \quad (3.1)$$

De maneira semelhante, como na Equação (3.2), quando se refere as linhas utiliza-se, por exemplo, $\mathbf{r}_j$ para indicar a j -ésima linha de $\mathbf{D}$.

$$\begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_j \\ \vdots \\ \mathbf{r}_N \end{bmatrix}_{N \times p} \begin{bmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_p \end{bmatrix}_{p \times 1} = \begin{bmatrix} t_1 \\ \vdots \\ t_j \\ \vdots \\ t_N \end{bmatrix}_{N \times 1} \quad (3.2)$$

Um problema existente ao se trabalhar com muitas variáveis é como identificar quais destas são realmente relevantes (ativas) ou quais não são (inativas). Os métodos de seleção de modelos são geralmente utilizados com esse propósito. Um grupo de técnicas, particularmente interessantes, para a redução de modelos são as técnicas gulosas, as quais permitem que soluções geralmente subótimas sejam obtidas de uma maneira mais fácil e rápida. Um algoritmo guloso faz uma escolha que aparentemente é a melhor, em determinado passo do algoritmo e, ao fim dos passos, é esperado obter uma solução próxima da ótima sendo que, de uma forma geral, esses algoritmos não fazem mudanças nas escolhas feitas anteriormente (HAMEED, 2012). As características citadas anteriormente geralmente

levam os algoritmos gulosos a ter um custo computacional baixo e, mesmo que levem a soluções subótimas, se tornam atrantes por proverem uma resposta em tempo adequado.

3.2 Seleção de Modelos

A obtenção de representações mais estáveis de um modelo e uma melhoria no desempenho de classificadores ou regressores podem ser alcançadas através da eliminação de algumas variáveis em um modelo linear. Na literatura de reconhecimento de padrões, os termos geralmente utilizados são seleção de características e extração de características (FUKUNAGA, 1990).

A redução de dimensionalidade pode ser necessária essencialmente por várias razões, por exemplo, permitir a visualização ou exploração das estruturas dos dados, remoção de ruídos, compressão ou extração de um espaço de características representativo. Geralmente os valores estimados com um modelo reduzido são mais precisos se baseados nas variáveis mais relevantes (CUNNINGHAM; GHAMRANI, 2015).

Seja $\mathbf{d}_j \in \mathbb{R}^p$, tal que j indica uma das N instâncias de dimensão p e $\mathbf{t}$ a variável dependente. A seleção de um subconjunto de variáveis A contido no conjunto das combinações possíveis das variáveis dependentes, o qual possui cardinalidade 2^p , deve ser feita de forma que o erro quadrático médio (*Mean Square Error*, MSE) seja minimizado, tal que

$$MSE = \frac{1}{m} \sum_{i=1}^m \|\mathbf{t} - \hat{\mathbf{t}}\|_2^2 \quad (3.3)$$

$$MSE = \frac{1}{m} \sum_{i=1}^m \|\mathbf{t} - \mathbf{D} \mathbf{z}\|_2^2 \quad (3.4)$$

em que $\hat{\mathbf{t}}$ é a saída obtida por

$$\mathbf{D} \mathbf{A} \mathbf{z} = \hat{\mathbf{t}} \approx \mathbf{t} \quad (3.5)$$

De uma forma geral, obter o vetor esparso $\mathbf{z}$, tal o qual apresentado na Equação (3.4), exige que se restrinja o número de elementos diferentes de zero na solução do sistema $\mathbf{z} = \mathbf{D}^\dagger \mathbf{t}$, em que $\mathbf{D}^\dagger$ é a pseudo inversa de Moore-Penrose. Considere k como o número máximo desses elementos e, assim, o problema se resume ao cômputo da seguinte expressão.

$$\min \|\mathbf{t} - \mathbf{D} \mathbf{z}\|_2 \text{ sujeito a } \|\mathbf{z}\|_0 \leq k \quad (3.6)$$

em que $\|\mathbf{z}\|_0$ é a norma L_0 a qual indica a quantidade de elementos diferentes de zero em $\mathbf{z}$, podendo ser computado também como:

$$\|\mathbf{z}\|_0 = \sum_{i=1}^N \Delta_i, \text{ tal que } \Delta_i = \begin{cases} 1, & \text{se } z_i \neq 0 \\ 0, & \text{caso contrário} \end{cases} \quad (3.7)$$

3.2.1 Seleção de Melhor Subconjunto

Uma ideia natural para impor esparsidade no vetor $\mathbf{z}$, é testar todos os subconjuntos possíveis de variáveis e escolher aquele que apresentar menor erro quadrático médio. Entretanto a escolha desse subconjunto é um problema NP-difícil (NATARAJAN, 1995), para o qual são necessários 2^p testes, tornando essa técnica de seleção de modelo rapidamente inviável de computar quando o valor de p se eleva. É comum a limitação da quantidade de variáveis utilizadas pela aplicação de um algoritmo de seleção de melhor subconjunto (*Best Subset Selection, BSS*). Isso reduz a ordem do problema consideravelmente, porém, seu custo ainda permanece fatorial. A quantidade de testes necessários para a seleção de um modelo limitado a k variáveis é dada por

$$C_k^p = \frac{p!}{k!(p-k)!} \quad (3.8)$$

É importante ressaltar que, para cada teste, um modelo com k variáveis deve ser estimado e, portanto, são necessárias várias soluções de sistemas lineares para a obtenção do modelo final. Atualmente, a utilização de tais algoritmos não são recomendados para uma quantidade de variáveis maiores que 30, caso no qual são necessários mais de 10^{32} testes. (Bertsimas; King; Mazumder, 2015; WEISBERG, 2005). Posteriormente, o algoritmo *leaps and bounds* foi capaz de reduzir o custo computacional consideravelmente ao utilizar técnicas de programação dinâmica para estimar os limites do erro das variáveis ainda não inseridas no subconjunto final (FURNIVAL, 1974).

3.2.2 Seleção Progressiva

Como dito na seção anterior, a técnica BSS tem um elevado custo computacional, em razão disso é inviável para conjunto de dados onde a quantidade de variáveis é elevada. Como alternativa para a obtenção de um subconjunto de variáveis de forma menos onerosa, pode-se fazer a seleção de variáveis sequencialmente. Essa seleção sequencial pode levar a um subconjunto não ótimo, mas, geralmente, com resultados aceitáveis na prática (WEISBERG, 2005).

A técnica de seleção progressiva (*Forward Selection, FS*) consiste em selecionar as variáveis de acordo com algum critério, como por exemplo, a correlação entre as variáveis e o resíduo estimado. É uma técnica simples e direta para estimar o modelo, que pode ser considerada muito gulosa, uma vez que pela escolha da variável mais correlacionada (ou outro critério escolhido a priori) pode-se ignorar outras possíveis variáveis (EFRON *et al.*, 2004a).

Ao se utilizar o cômputo da correlação como critério de seleção a técnica de seleção progressiva é equivalente à busca por correspondência (*Matching Pursuit, MP*), bastante conhecida na comunidade de Processamento de Sinais (MALLAT, 2008; MALLAT;

ZHANG, 1993). Além da escolha do critério de seleção, geralmente outras duas definições são impostas ao MP, a saber: a quantidade máxima de variáveis k que entram no modelo e o erro máximo aceitável s . A técnica é simples e intuitiva, porém, por ser muito gulosa, converge para um modelo esparsos de forma mais lenta. Em outras palavras, a medida que novas variáveis são selecionadas, o erro quadrático médio diminui mais lentamente que outras técnicas (STURM; CHRISTENSEN, 2010).

Algoritmo 1 Matching Pursuit - Calcula $\mathbf{z}$ de forma esparsa limitado a k variáveis

Entrada:

$\mathbf{D}$, Características dos padrões de treinamento
 $\mathbf{t}$, Rótulo dos padrões de treinamento
 k , Número máximo de variáveis
 s , Erro máximo aceitável

Saída:

$\mathbf{z}$, Vetor de pesos
 I , Conjunto contendo as variáveis escolhidas

```

1:  $i \leftarrow 1$ 
2:  $\delta_i \leftarrow \mathbf{t}$ 
3:  $I \leftarrow \{\}$ 
4:  $k \leftarrow \min(k, p)$  onde  $p$  é a quantidade de variáveis em  $\mathbf{D}$ 
5:  $A \leftarrow \{1 \dots p\}$ 
6: Enquanto  $\|\delta_i\| > s$  e  $i \leq k$  faça
7:    $\mathbf{c} \leftarrow \mathbf{D}_A^T \delta_i$ 
8:    $j \leftarrow \operatorname{argmax}_j \{|\mathbf{c}_j|\}$ 
9:    $I \leftarrow I \cup \{j\}$ 
10:   $\mathbf{z}_i \leftarrow \mathbf{c}_j$ 
11:   $\delta_{i+1} \leftarrow \delta_i - \mathbf{d}_j \mathbf{z}_i$ 
12:   $i \leftarrow i + 1$ 
13:   $A \leftarrow A - \{j\}$ 
14: Fim Enquanto

```

No Algoritmo 1 é descrita a técnica de forma mais detalhada. Na linha 2 o resíduo é iniciado com o valor da variável dependente, na linha 4 é verificada a quantidade máxima de variáveis que será utilizada no modelo final, enquanto na linha 5, A é iniciado com o índice de todas as variáveis. Dentro do laço, na linha 7 é verificada a correlação do resíduo com as variáveis independentes que estão no conjunto A . Após a escolha da mais correlacionada, na linha 11 é feita a atualização do resíduo. Posteriormente o conjunto A é atualizado (linha 8). O laço se repete até atingir a quantidade máxima de variáveis e o erro estar abaixo do aceitável.

Do ponto de vista dos mínimos quadrados o vetor $\mathbf{z}$, calculado pela técnica MP, provê uma solução próxima da ótima, porém a fim de aprimorar os resultados foi proposta a técnica busca por correspondência ortogonal (*Orthogonal Matching Pursuit*, OMP), na qual o vetor $\mathbf{z}$ é recalculado utilizando todos os índices escolhidos até então, ao contrário

da técnica MP na qual cada z_i é calculado de forma independente (CAI; WANG, 2011; HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

Recalcular o vetor $\mathbf{z}$ utilizando todas as variáveis escolhidas até então, é equivalente a considerar que essas mesmas variáveis formam uma base ortogonal, daí surge a denominação. Esta técnica tem sido muito aplicada em recuperação e aproximação de sinais e resultados obtidos por Cai e Wang (CAI; WANG, 2011) mostram que os erros obtidos são relativamente pequenos. Detalhes para o método OMP são apresentados no Algoritmo 2. No algoritmo nota-se a grande semelhança entre o MP e OMP. Na linha 10 o OMP estima o vetor de pesos $\mathbf{z}$ utilizando todas as amostras escolhidas, as quais estão no conjunto I . Na linha 11 o novo resíduo é calculado, desta vez, em relação ao vetor de rótulos do modelo.

Algoritmo 2 Orthogonal Matching Pursuit - Calcula $\mathbf{z}$ de forma esparsa limitado a k variáveis

Entrada:

$\mathbf{D}$, Características dos padrões de treinamento
 $\mathbf{t}$, Rótulo dos padrões de treinamento
 k , Número máximo de variáveis
 s , Erro máximo aceitável

Saída:

$\mathbf{z}$, Vetor de pesos
 I , Conjunto contendo as variáveis escolhidas

```

1:  $i \leftarrow 1$ 
2:  $\boldsymbol{\delta}_i \leftarrow \mathbf{t}$ 
3:  $k \leftarrow \min(k, p)$  onde  $p$  é a quantidade de variáveis  $\mathbf{D}$ 
4:  $A \leftarrow \{1 \dots p\}$ 
5:  $I \leftarrow \{\}$ 
6: Enquanto  $\|\boldsymbol{\delta}_i\| > s$  e  $i \leq k$  faça
7:    $\mathbf{c} \leftarrow \mathbf{D}_I^T \boldsymbol{\delta}_i$ 
8:    $j \leftarrow \operatorname{argmax} \{|\mathbf{c}|\}$ 
9:    $I \leftarrow I \cup \{j\}$ 
10:   $\mathbf{z} \leftarrow \mathbf{D}_I^\dagger \boldsymbol{\delta}_i$ 
11:   $\boldsymbol{\delta}_{i+1} \leftarrow \mathbf{t} - \mathbf{D}_I \mathbf{z}$ 
12:   $i \leftarrow i + 1$ 
13:   $A \leftarrow A - \{j\}$ 
14: Fim Enquanto

```

3.2.3 Seleção Progressiva em Estágios

As técnicas *Matching Pursuit* e *Forward Selection* tendem a ser muito gulosas, isto é, a heurística escolhida tende a ser muito otimista e geralmente não resulta em modelos esparsos e com baixo erro quadrático médio. A fim de fazer uma seleção das variáveis de forma mais cautelosa, foi proposta a técnica de seleção progressiva em estágios (*Forward Stagewise Selection*, *FSS*). Nesta técnica se escolhe a variável mais correlacionada com o

resíduo atual, porém, a solução do sistema é atualizada em pequenos passos e, devido a isto, outras componentes em determinada iteração podem ter uma maior correlação com o este resíduo.

A técnica FSS é iniciada de maneira similar ao MP, com todos os coeficientes iguais a zero e o resíduo atual igual a variável dependente. Em cada passo da iteração é escolhida a j -ésima variável mais correlacionada com o resíduo, a solução $\mathbf{z}$ é atualizada em direção de $\mathbf{d}_j$ por um fator pequeno B que indica o tamanho do passo. Uma nova variável só é escolhida quando outra variável possuir uma maior correlação com o resíduo atual. No Algoritmo 3 é apresentado em detalhes o cálculo da solução esparsa $\mathbf{z}$. De maneira muito similar ao FS, o *Forward Stagewise Selection* estima o peso z_i em direção a variável mais correlacionada, porém, desta vez, apenas uma parcela B será utilizada, como pode ser vista nas linha 11 e 12. Além disso, é adicionado mais um critério de parada que limita o número de passos que podem ser realizados (linha 7).

Algoritmo 3 Forward Stagewise Selection - Calcula $\mathbf{z}$ de forma esparsa limitado a k variáveis

Entrada:

$\mathbf{D}$, Características dos padrões de treinamento
 $\mathbf{t}$, Rótulo dos padrões de treinamento
 k , Número máximo de variáveis
 $maxIter$, Número máximo de iterações
 B , Tamanho do passo de atualização

Saída:

$\mathbf{z}$, Solução do sistema
 I , Índices das variáveis escolhidas

```

1:  $i \leftarrow 1$ 
2:  $it \leftarrow 1$ 
3:  $\delta_i \leftarrow \mathbf{t}$ 
4:  $k \leftarrow \min(k, p)$ 
5:  $ultIndice \leftarrow \emptyset$ 
6:  $\mathbf{z} \leftarrow \mathbf{0}$ 
7: Enquanto  $\|\delta_i\| > s$  e  $i \leq k$  e  $it \leq maxIter$  faça
8:    $\mathbf{c} \leftarrow \mathbf{D}_A^T \mathbf{r}_i$ 
9:    $j \leftarrow \arg\max\{|\mathbf{c}_j|\}$ 
10:   $I \leftarrow I \cup \{j\}$ 
11:   $\mathbf{z}_i \leftarrow \mathbf{z}_i + B \sin(\mathbf{c}_j)$ 
12:   $\delta_{i+1} \leftarrow \mathbf{t} - B \sin(\mathbf{c}_j) \mathbf{d}_i$ 
13:   $it \leftarrow it + 1$ 
14:  se  $j = ultIndice$  então
15:     $i \leftarrow i + 1$ 
16:     $A \leftarrow A - \{j\}$ 
17:     $ultIndice \leftarrow j$ 
18:  Fim se
19: Fim Enquanto

```

3.2.4 Eliminação para Trás

Como dito na seção anterior, na técnica *Forward Selection* a seleção é feita a partir de um conjunto vazio de variáveis que cresce progressivamente, geralmente melhorando o desempenho. Na técnica eliminação para trás (*Backward Elimination, BE*), também conhecida como (*Backward Selection, BS*), todo o conjunto de variáveis é selecionado *a priori*, depois disso, algum método deve indicar quais variáveis (uma ou mais) são menos relevantes, para então serem removidas. Considerando um sistema $D\mathbf{z} = \mathbf{t}$, a escolha por qual variável será removida pode ser feita avaliando o valor $|z_i|$, obtido a partir da solução do sistema. Valores de $|z_i|$ próximos de 0, indicam que as variáveis não possuem uma contribuição significativa e, portanto, podem ser removidas. Esse processo pode ser feito, a partir das variáveis restantes, até que um critério de parada seja atingido (GUYON STEVE GUNN, 2006).

Uma vez que a escolha é feita a partir de um modelo treinado com todas as variáveis, essa técnica possui uma grande desvantagem em conjuntos de dados com uma grande dimensionalidade e/ou com dimensões correlacionadas. Nesses conjuntos as iterações iniciais demandam um maior recurso computacional e caso o modelo desejado seja muito pequeno, em relação à quantidade de variáveis, muito desse recurso será utilizado eliminando pequenas quantidades de variáveis. No Algoritmo 4 é apresentada a técnica *Backward Selection* levando em consideração, como critério, o valor dos pesos obtidos. Como pode ser visto, na linha 1 é estimado o vetor de pesos utilizando todas as variáveis e, na linha 3, o resíduo obtido com essa estimativa. Na linha 8 é removido do conjunto de variáveis do modelo aquela que o peso associado possui menor valor. O resíduo atual do modelo é estimado nas linhas seguintes (linhas 9 e 10) e isso se repete até que algum dos critérios de parada sejam alcançados.

3.3 Máquinas de Vetores Suporte de Mínimos Quadráticos Otimamente Podadas (Proposta)

Nesta seção é apresentada a proposta, denominada OP-LSSVM, a qual é uma técnica que impõe esparsidade nas máquinas de vetores suporte de mínimos quadrados utilizando a técnica de regressão esparsa multiresposta (*Multiresponse Sparse Regression, MRSR*) como método de seleção de vetores suporte. A fim de reforçar a generalização, o *Leave-one-out* é utilizado para escolher o melhor subconjunto de vetores. O algoritmo pode ser dividido em três grandes passos. O primeiro é construir a matriz D , conforme as equações (2.45) ou (2.57) a partir de todos os dados de treinamento, levando em consideração se o problema é de classificação ou de regressão. Após isso, a técnica MRSR é aplicada para estimar um modelo $\mathbf{z}$ restrito a no máximo k variáveis, neste contexto vetores suporte. Finalmente uma técnica de validação cruzada é utilizada para restringir

Algoritmo 4 Backward Selection - Calcula $\mathbf{z}$ de forma esparsa limitado a k variáveis**Entrada:**

$\mathbf{D}$, Características dos padrões de treinamento
 $\mathbf{t}$, Rótulo dos padrões de treinamento
 k , Número máximo de variáveis
 s , Erro máximo aceitável

Saída:

$\mathbf{z}$, Vetor de pesos
 ind , Conjunto contendo as variáveis escolhidas

```

1:  $\mathbf{z} \leftarrow \mathbf{D}^\dagger \mathbf{t}$ 
2:  $i \leftarrow p$ 
3:  $\boldsymbol{\delta}_i \leftarrow \mathbf{t} - \mathbf{zD}$ 
4:  $k \leftarrow \min(k, p)$  onde  $p$  é a dimensionalidade de  $\mathbf{D}$ 
5:  $I \leftarrow \{1 \dots p\}$ 
6: Enquanto  $\|\boldsymbol{\delta}_i\| > s$  e  $i \geq k$  faça
7:    $j \leftarrow \operatorname{argmin}_j \{|\mathbf{z}_j|\}$ 
8:    $I \leftarrow I - \{j\}$ 
9:    $\mathbf{z} \leftarrow \mathbf{D}_I^\dagger \mathbf{t}$ 
10:   $\boldsymbol{\delta}_{i+1} \leftarrow \mathbf{t} - \mathbf{D}_I \mathbf{z}$ 
11:   $i \leftarrow i - 1$ 
12: Fim Enquanto
  
```

ainda mais a quantidade de vetores suporte, ou seja, buscar mais esparsidade.

Na Figura 7 é apresentado um diagrama de blocos que exhibe os principais passos para treinamento de uma OP-LSSVM.

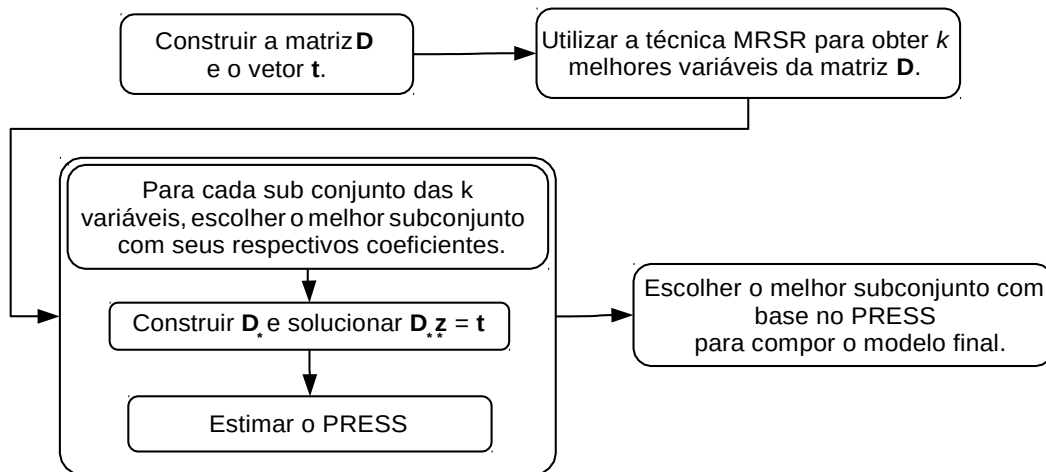


Figura 7 – Fluxograma do treinamento da OP-LSSVM.

O processo inicia-se no bloco superior esquerdo, onde todos os padrões são utilizadas para contruir a matriz $\mathbf{D}$ e o vetor $\mathbf{y}$. A solução $\mathbf{z}$ pode ser estimada pela inversão da matriz da mesma forma que se faz para uma LSSVM. Após isso, o vetor $\mathbf{z}$ é estimado utilizando

a técnica MRSR, a qual provê uma ordenação dos padrões em termos de relevância. O bloco inferior esquerdo representa o processo de escolha do melhor subconjunto de vetores suporte. Nela $\mathbf{D}_*$ é a matriz de *kernel* formada pelas k primeiras variáveis escolhidas. Finalmente, no último bloco, o PRESS (*Predicted Residual Error Sum of Squares*), uma técnica de validação cruzada equivalente ao *leave-one-out* (LOO), é utilizado para avaliar e selecionar o melhor subconjunto de vetores suporte.

3.3.1 Seleção através da Regressão Esparsa Multiresposta

Na regressão esparsa multiresposta (*Multiresponse Sparse Regression*, MRSR) (SIMILÄ; TIKKA, 2005) as variáveis são selecionadas de forma sequencial e se configura como uma técnica que estende o *Least Angle Regression* (LARS) (EFRON *et al.*, 2004b). O *Least Angle Regression* e, por consequência a técnica MRSR, é uma versão mais eficiente da técnica *Forward Stagewise Selection*. A melhoria do método é alcançada pelo cálculo automático do passo B em cada iteração da técnica. Além disso ao invés de realizar várias iterações até surgir uma variável com igual correlação, nela se calcula a direção e o tamanho máximo do passo que pode ser dado até que mais de uma variável possua a mesma correlação. Dessa forma, o número de passos necessários para obter todo o conjunto de soluções é o próprio tamanho do conjunto de treinamento. A técnica MRSR, por ser multiresposta, é aplicada em sistema lineares do tipo

$$\mathbf{D}\mathbf{Z} = \mathbf{T} \quad (3.9)$$

em que $\mathbf{Z}$ e $\mathbf{T}$ são matrizes. A fim de compatibilizar sua descrição com a proposição em termos de uma LSSVM será apresentada a sua versão com uma única resposta, ou seja, com $\mathbf{Z}$ e $\mathbf{T}$ reduzidos aos vetores $\mathbf{z}$ e $\mathbf{t}$, tal como no seguinte sistema linear

$$\mathbf{t}^k = \mathbf{D}\mathbf{z}^k \quad (3.10)$$

Assim considere $\mathbf{t}^k \in \mathbb{R}^N$ um vetor que contém as variáveis dependentes associadas à uma variável $\mathbf{d}_i \in \mathbb{R}^N$. A cada k iteração do MRSR, uma variável é adicionada ao conjunto de variáveis ativas. O vetor $\mathbf{z}^k$ contém os pesos utilizados para estimar a aproximação $\mathbf{t}^k$ sendo que na k -ésima iteração há k variáveis com valores diferentes de zero. Isso, em termos práticos, significa que apenas k variáveis de $\mathbf{D}$ são efetivamente utilizadas e, assim, as variáveis iguais a zero podem ser removidas sem impactar no desempenho do modelo. É importante ressaltar que, para uma OP-LSSVM, o vetor $\mathbf{z}$ apresenta a solução e consequentemente os multiplicadores de Lagrange, a matrix $\mathbf{D}$, quadrada de tamanho $N + 1$, representa a matriz que contém *kernel* utilizado para estimar os rótulos $\mathbf{t}$.

No início, quando $k = 0$, todos os elementos dos vetores $\mathbf{t}$ e $\mathbf{z}$ são iniciados com zero. A escolha da variável que deve entrar no conjunto é feita com base na correlação

cumulativa absoluta entre as variáveis independentes $\mathbf{D}$ e o resíduo acumulado, definido como $\mathbf{t} - \mathbf{t}^k$. Assim tem-se que, em determinada iteração k , a correlação da j -ésima variável contida na matriz $\mathbf{D}$ com o resíduo é dada por:

$$c_j^k = ||(\mathbf{t} - \mathbf{t}^k)^\top \mathbf{d}_j||_1 \quad (3.11)$$

A norma apresentada na Equação (3.11) permite que a correlação entre a j -ésima variável $\mathbf{d}_j$ e o resíduo acumulado seja resumida a um valor. Com base em tais valores, pode-se obter a variável mais correlacionada por

$$c_{max}^k = \max_j \{c_j^k\}, A = \{j | c_j^k = c_{max}^k\}. \quad (3.12)$$

em que A é o conjunto de variáveis escolhidas na iteração k . Com base apenas nas variáveis já adicionadas pode-se recalculer a resposta $\bar{\mathbf{t}}^{k+1}$ utilizando o modelo estimado $\bar{\mathbf{z}}^{k+1}$ da seguinte forma

$$\begin{aligned} \bar{\mathbf{z}}^{k+1} &= (\mathbf{D}_A^\top \mathbf{D}_A)^{-1} \mathbf{D}_A^\top \mathbf{t} \\ \bar{\mathbf{t}}^{k+1} &= \mathbf{D}_A \bar{\mathbf{z}}^{k+1}. \end{aligned} \quad (3.13)$$

A técnica constrói de forma iterativa um modelo para regressão linear de uma função através de sucessivas e pequenas atualizações na solução $\mathbf{z}^k$. O tamanho dessa atualização, aqui chamado de passo, é definido por B tal que $0 < B < |c_j|$ em que c_j é a correlação entre a variável dependente e a variável independente j , por vezes também chamado de regressor e denotado por $\mathbf{d}_j$. Neste contexto j é o índice do regressor com maior correlação absoluta.

Ao considerar o valor do passo $B = |c_j|$, leva a técnica a se comportar de maneira idêntica a técnica *Forward Selection*. Por outro lado, o uso de um B muito pequeno, faz com que a técnica demore a convergir. A fim de poupar recursos computacionais, tanto o LARS quanto o MRSR estimam o tamanho do passo e a direção em que a solução deve ser atualizada, fazendo esse ajuste de uma só vez e, assim, com um menor custo computacional. Um ponto chave, durante a escolha da direção e do passo a serem seguidos, é que as variáveis adicionadas ao conjunto A devem sempre possuir igual correlação com todas as dimensões de $\mathbf{t}$. Assim pode-se definir a nova regra de atualização da solução em função do passo B^k .

$$\mathbf{z}^{k+1} = (1 - B^k) \mathbf{z}^k + B^k \bar{\mathbf{z}}^{k+1} \quad (3.14)$$

e, consequentemente, tem-se

$$\mathbf{t}^{k+1} = (1 - B^k) \mathbf{t}^k + B^k \bar{\mathbf{t}}^{k+1} \quad (3.15)$$

O vetor $\bar{\mathbf{z}}^{k+1}$, apresentado na Equação (3.13), indica em que direção a nova solução irá, quando uma nova variável é adicionada ao conjunto A. Enquanto o passo B faz um controle dessa direção, garantindo que essa não altere a correlação das outras variáveis. A fim de manter essa última restrição o valor do passo

$$B^k = \min\{B | B^k \geq 0 \text{ e } B \in \Gamma_k \text{ para algum } j \notin A\}, \quad (3.16)$$

deve ser o menor valor positivo presente no conjunto

$$\Gamma_k = \frac{c_{\max}^k + s^T(\mathbf{t} - \mathbf{t}^k)^T \mathbf{d}_j}{c_{\max}^k + s^T(\bar{\mathbf{t}}^{k+1} - \mathbf{t}^k)^T \mathbf{d}_j} \quad s \in S \quad (3.17)$$

em que S é o conjunto formado por todos os 2^p vetores sinal de tamanho $p \times 1$. Abaixo é apresentado alguns dos conjuntos possíveis S para p igual a 1, 2 e 3 respectivamente.

$$\begin{aligned} S_1 &= \{(1), (-1)\} \\ S_2 &= \{(1, 1), (1, -1), (-1, 1), (-1, -1)\} \\ S_3 &= \{(1, 1, 1), (1, 1, -1), (1, -1, 1), (1, -1, -1), (-1, 1, 1), \\ &\quad (-1, 1, -1), (-1, -1, 1), (-1, -1, -1)\} \end{aligned} \quad (3.18)$$

Ressalta-se que para a proposta, que se baseia no LSSVM, utiliza-se o conjunto S_1 . Após estimar o valor de $\mathbf{z}^{k+1}$, é iniciada a próxima iteração. O funcionamento da técnica MRSR é apresentado em detalhes no Algoritmo 5.

No contexto da OP-LSSVM, a técnica MRSR provê a ordem de relevância das variáveis (vetores suporte) para composição do modelo, logo, tal ordem também indica a importância destas para a estimativa da variável dependente (rótulos). A fim de obter esparsidade em LSSVMs, outras técnicas para obtenção de modelos esparsos podem ser utilizadas, tais como as apresentadas anteriormente nesse capítulo. Para fins de comparação com nossa proposta foi utilizada uma versão similar a uma já encontrada na literatura que utiliza busca por correspondência (YANG; BOUZERDOUM; PHUNG, 2010). Assim, para uma comparação justa também foi adicionada uma etapa extra pela utilização do PRESS. Neste trabalho esta técnica é denominada MP-LSSVM.

3.3.2 Predição da Soma dos Erros Quadrados (PRESS)

Com a intenção de reforçar a generalização das LSSVMs e obter um subconjunto ainda menor de vetores suporte, uma técnica de validação cruzada, tal como o *leave-one-out*, pode ser aplicada. Esta é utilizada no terceiro e último passo para o treinamento da OP-LSSVM.

Validação cruzada (*cross validation*) é uma técnica utilizada para avaliar como o modelo treinado se comporta quando testado com amostras não utilizadas no processo de obtenção do modelo. A maneira mais simples de se realizar uma avaliação do desempenho

Algoritmo 5 MRSR - Calcula $\mathbf{z}$ de forma esparsa limitado a k variáveis**Entrada:**

$\mathbf{D}$, uma matriz (m, n)
 $\mathbf{t}$, um vetor de tamanho m
 $kmax$, número máximo de variáveis

Saída:

$\mathbf{z}$, solução
 $\mathbf{A}$, um vetor com os índices escolhidos

```

1:  $k \leftarrow 1$ 
2:  $\mathbf{z}^k \leftarrow \mathbf{0}$ 
3:  $\mathbf{t}^k \leftarrow \mathbf{0}$ 
4:  $\mathbf{A} \leftarrow \emptyset$ 
5: Enquanto  $k < kmax$  faça
6:    $c_j^k = \|(\mathbf{t} - \mathbf{t}^k)^\top \mathbf{d}_j\|_1$ 
7:    $j \leftarrow \operatorname{argmax}_j \{c_j^k\}$ 
8:    $c_{max}^k = c_j^k$ 
9:    $\mathbf{A} = \mathbf{A} \cup j$ 
10:   $\bar{\mathbf{z}}^{k+1} = (\mathbf{D}^\top \mathbf{D}_{\mathbf{A}})^{-1} \mathbf{D}_{\mathbf{A}}^\top \mathbf{t}$ 
11:   $\bar{\mathbf{t}}^{k+1} = \mathbf{D}_{\mathbf{A}} \bar{\mathbf{z}}^{k+1}$ 
12:   $a = c_{max}^k + \mathbf{s}_p^\top (\mathbf{t} - \mathbf{t}^k)^\top \mathbf{d}_j$ 
13:   $b = c_{max}^k + \mathbf{s}_p^\top (\bar{\mathbf{t}}^{k+1} - \mathbf{t}^k)^\top \mathbf{d}_j$ 
14:   $\Gamma_k = a/b$ 
15:   $\mathbf{B}^k = \min\{\mathbf{B} | \mathbf{B}^k \geq 0 \text{ e } \mathbf{B} \in \Gamma_k \text{ para algum } j \notin \mathbf{A}\}$ 
16:   $\mathbf{z}^{k+1} = (1 - \mathbf{B}^k) \mathbf{z}^k + \mathbf{B}^k \bar{\mathbf{z}}^{k+1}$ 
17:   $\mathbf{t}^{k+1} = (1 - \mathbf{B}^k) \mathbf{t} + \mathbf{B}^k \bar{\mathbf{t}}^{k+1}$ 
18:   $k = k + 1$ 
19: Fim Enquanto

```

de um modelo é dividir o conjunto de dados em dois, um para treinamento e outro para teste. Este último, independente do primeiro, é utilizado para avaliar a qualidade do treinamento, o qual, geralmente, é dada pelo cômputo do erro quadrático médio.

A forma como o conjunto é dividido define o nome da técnica de validação cruzada como por exemplo, para o caso no qual o conjunto de dados é dividido em 10 partições, denomina-se *10-fold cross validation*. Como dito anteriormente, pode-se utilizar uma quantidade variada de padrões para realizar a avaliação do modelo. Quando o número de partições é a máxima possível e, portanto, igual a quantidade de instâncias no conjunto de dados, a técnica de validação cruzada recebe o nome de *leave-one-out*.

Na Equação (3.19) é apresentado o cômputo do erro *leave-one-out*, E_{loo} obtido ao avaliar um conjunto de dados. Nessa equação, $\hat{t}_j$ é o rótulo estimado para um padrão $\mathbf{r}_j$ por um modelo quando todos os padrões, com exceção do j -ésimo, são utilizados para treiná-lo, enquanto t_j é o rótulo esperado para tal padrão.

$$\square \quad \hat{E} = \frac{1}{N} \sum_{j=1}^N (t_j - \hat{t}_j)^2 \quad (3.19)$$

Algoritmo 6 Cômputo do Erro *leave-one-out***Entrada:** $\mathbf{D}$, Conjunto de treinamento $\mathbf{t}$, Rótulo dos padrões de treinamento**Saída:** E_{loo} , Erro *leave-one-out*

- 1: $j \leftarrow N$, Padrão escolhido para ser removido
- 2: $U \leftarrow \emptyset$, Conjunto dos índices dos padrões em uso
- 3: $E_{loo} \leftarrow 0$
- 4: **Enquanto** $j < N$ **faça**
- 5: $U \leftarrow \{1 \dots N\} - \{j\}$
- 6: $\mathbf{z}_{-j} \leftarrow \mathbf{D}_U^+ \mathbf{y}_U$
- 7: $e_{loo} = (\mathbf{t}_j - \mathbf{z}_{-j})^2$
- 8: $E_{loo} \leftarrow E_{loo} + e_{loo}$
- 9: $j \leftarrow j - 1$
- 10: **Fim Enquanto**

Para um conjunto de treinamento $\mathbf{D}$, o erro do *leave-one-out* pode ser obtido como descrito no Algoritmo 6. Na linha 5, o conjunto U é formado pela remoção do j -ésimo índice que representa um dos padrões de treinamento. Na linha seguinte, o modelo $\mathbf{z}_{-j}$ é obtido utilizando os $N - 1$ padrões restantes do conjunto de treinamento. Na linha 7 o erro *leave-one-out*, quando o padrão j é removido, é estimado. Na linha 8 a soma dos erros quadrados é acumulado para os N padrões de treinamento.

Apesar de ser interessante a utilização do LOO como descrito na Equação (3.19), sabe-se que o custo computacional é elevado. Uma técnica mais elegante e direta apresentada por (ALLEN, 1974; TARPEY, 2000; MYERS, 1990) permite estimar o valor do PRESS com o auxílio da matriz chapéu.

Durante a estimativa de um modelo linear utilizando os mínimos quadrados é muito comum e útil determinar qual a influência de cada padrão no modelo estimado. Essa informação pode ser obtida à partir da matriz chapéu (*hat matrix*). O termo matriz chapéu surge da adição de um (^) para representar o resultado estimado de um modelo quando os próprios dados de treinamento são utilizados (HOAGLIN; WELSCH, 1978). Seja $\mathbf{D}$ uma matriz ($N \times p$) que representa o conjunto de dados a matriz chapéu $\mathbf{H}$ pode ser computada por

$$\mathbf{H} = \mathbf{D}(\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \quad (3.20)$$

A expressão apresentada na Equação (3.20) surge do seguinte cômputo

$$\begin{aligned}
 \hat{\mathbf{t}} &= \mathbf{D} \mathbf{z} \\
 \mathbf{z} &= \mathbf{D}^+ \mathbf{t} \\
 \mathbf{z} &= (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{t} \\
 \hat{\mathbf{t}} &= \mathbf{D}(\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{t}
 \end{aligned} \quad (3.21)$$

$\underbrace{\hspace{10em}}_{\mathbf{H}} \quad \underbrace{\hspace{10em}}_{\mathbf{x}}$

de tal maneira que a matriz $\mathbf{H}$ é quadrada de tamanho N . A diagonal principal da matriz chapéu possui uma valiosa informação sobre as variáveis independentes, da qual é possível identificar os pontos que estão mais distantes dos dados como um todo. Isso é importante, pois permite determinar a localização de amostras discrepantes (*outliers*) ou a influência que um determinado padrão tem sobre o modelo linear calculado (FOUSSIER, 2006). O valor da diagonal principal para o j -ésimo padrão pode ser computado da seguinte forma, a saber:

$$h_{jj} = \mathbf{r}_j(\mathbf{D}^T \mathbf{D})^{-1} \mathbf{r}_j^T \quad (3.22)$$

Levando em consideração a equação anterior, o erro de um modelo, de acordo com Myers (1990) e Miche *et al.* (2010) dada a remoção do padrão $\mathbf{r}_j$ pode ser computado como o mostrado a seguir.

$$e_{loo}^j = \frac{t_j - \hat{t}_j}{1 - h_{jj}}, \quad (3.23)$$

em que h_{jj} refere-se a diagonal principal da matriz chapéu e $\hat{t}_j$ é o valor estimado de $\mathbf{r}_j$ utilizando o vetor $\mathbf{z}$ apresentado na Equação (3.21). Dessa forma, o *PRESS* é o resultado da soma de todos os erros provocados pela remoção de um padrão $\mathbf{r}_j$ e pode ser obtido por

$$E = PRESS = \sum_{j=1}^N (e_{loo}^j)^2. \quad (3.24)$$

Uma vez que o cômputo do *leave-one-out* é computacionalmente intenso o *PRESS* é uma alternativa promissora que permite este cálculo de forma mais eficiente, direta e, conseqüentemente, com um menor custo computacional. Mais detalhes sobre o *PRESS* e sua equivalência ao LOO podem ser verificados no Apêndice A. No Algoritmo 7 é apresentado o cômputo do *PRESS* em detalhes. Na linha 1, tem-se o cálculo da matriz chapéu. Na linha 2 é realizada a estimativa dos padrões de treinamento com a utilização da matriz chapéu. Na linha 4, a função diagonal retorna um vetor com todos os elementos da diagonal principal da matriz $\mathbf{H}$, a qual é subtraída do vetor de uns, $\mathbf{1}$, para a estimativa do vetor auxiliar $\mathbf{b}$. Finalmente, na linha 5, é obtida a estimativa do *PRESS*. Com o algoritmo apresentado, pode-se notar a economia, em termos de custo computacional, da utilização do *PRESS*, frente a formulação do *leave-one-out*.

3.3.3 Construção do Modelo Final

Levando em consideração a ordem de relevância dos vetores escolhidos pelo MRSR, uma busca por cada subconjunto formado pelos k primeiros vetores suporte é realizada com base no *PRESS*. Feito isto, o melhor subconjunto é utilizado para compor o modelo final. O Algoritmo 8 ilustra o processo de seleção pelo LOO. O laço apresentado na linha 2 ocorre para todos os valores de k no intervalo de $[2, N+1]$. A seleção não ocorre a partir

Algoritmo 7 Cômputo do PRESS

Entrada:

$\mathbf{D}$, Conjunto de treinamento
 $\mathbf{t}$, Rótulo dos padrões de treinamento

Saída:

$PRESS$, Estimativa do PRESS
 1: $\mathbf{H} = \mathbf{D}(\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}^\top$
 2: $\hat{\mathbf{t}} = \mathbf{H}\mathbf{t}$
 3: $\mathbf{a} = \mathbf{t} - \hat{\mathbf{t}}$
 4: $\mathbf{b} = \mathbf{1} - \text{diag}(\mathbf{H})$
 5: $PRESS = \sum_{i=1}^N (a_i / b_i)^2$

do primeiro elemento de V pois, o primeiro elemento deste é o índice da primeira coluna da matriz $\mathbf{D}$, a qual possui informações relativas às restrições do problema dual da LSSVM. Na linha 3, o conjunto W_k recebe os índices dos k primeiros vetores ordenados por grau de relevância pela técnica MRSR. Na linha seguinte, o PRESS é calculado utilizando os vetores associados aos índices presentes em W_k . Nas linhas 6 e 7, o melhor subconjunto de vetores é escolhido.

Após a seleção do melhor subconjunto de vetores suporte através do *leave-one-out*, o modelo final é construído, em outras palavras, os multiplicadores de Lagrange e o intercepto são calculados. É importante ressaltar que a solução do sistema linear $\mathbf{D}\mathbf{z} = \mathbf{t}$ é utilizada tanto para a versão da OP-LSSVM para classificação quanto para a versão de regressão. As abordagens se distinguem basicamente em como a matriz $\mathbf{D}$ e o vetor $\mathbf{t}$ são formados. Quando a OP-LSSVM é utilizada para resolver problemas de classificação, têm-se o sistema apresentado na Equação (2.45) e é chamada OP-LSSVC, quando aplicado à problemas de regressão, têm-se o sistema apresentado na Equação (2.57) e é chamada OP-LSSVR. De maneira semelhante, quando utilizado o *Mathing Pursuit* como técnica de

Algoritmo 8 Seleção do melhor subconjunto pelo PRESS

Entrada:

$\mathbf{D}$, Matriz de kernel inicialmente criada a partir de todos os dados
 $\mathbf{t}$, Vetor de rótulos
 V , Conjunto com índices dos vetores ordenados pelo MRSR

Saída:

V^* , Subconjunto ótimo dos índices dos vetores suporte
 1: $k \leftarrow 2$
 2: **Enquanto** $k \leq N + 1$ **faça**
 3: $W_k \leftarrow V_{1...k}$
 4: $E_{loo}^k \leftarrow PRESS(D_{W_k}, \mathbf{t})$
 5: $k \leftarrow k + 1$
 6: **Fim Enquanto**
 7: $k^* \leftarrow \text{argmin}_k \{E_{loo}^k\}$
 8: $V^* \leftarrow V_{1...k^*}$

ordenação de vetores suporte, têm-se a MP-LSSVC para problemas de classificação e a MP-LSSVR para problemas de regressão.

No Algoritmo 9, o pseudo-código para as OP-LSSVMs é apresentado. Na linha 1 é realizada a ordenação dos vetores na matriz $\mathbf{D}$. O resultado desta ordenação é armazenado no vetor $\mathbf{r}$, o qual é utilizado em conjunto com a matriz $\mathbf{D}$ para obter o melhor subconjunto $\mathbf{s}$ de vetores através do PRESS (linhas 2 e 3). De posse da matriz $\mathbf{D}_*$, representando a matriz reduzida a qual faz uso apenas dos vetores presentes em $\mathbf{s}$, na linha 4, os multiplicadores de Lagrange e o intercepto para serem utilizados nas equações (2.48) e (2.59) para problemas de classificação e regressão respectivamente.

Algoritmo 9 OP-LSSVM

Entrada:

$\mathbf{D}$, Matriz inicialmente criada a partir de todos os dados

$\mathbf{t}$, Rótulos dos vetores suporte

Saída:

$\mathbf{z}$, Multiplicadores de Lagrange e intercepto

- | | |
|--|----------------------------|
| 1: $\mathbf{V} \leftarrow \text{MRSR}(\mathbf{D}, \mathbf{t})$ | #Ranqueamento das colunas |
| 2: $\mathbf{V}^* \leftarrow \text{Seleção pelo PRESS}(\mathbf{D}, \mathbf{t}, \mathbf{V})$ | #Subconjunto escolhido |
| 3: $\mathbf{D}_* \leftarrow \mathbf{D}_{\mathbf{V}^*}$ | #Construção do novo modelo |
| 4: $\mathbf{z}^* = (\mathbf{D}_*^T \mathbf{D}_*)^{-1} \mathbf{D}_*^T \mathbf{t}$ | #Solução do modelo |

* *

3.3.4 Conclusão

Neste capítulo foram apresentadas técnicas gulosas para a obtenção de modelos lineares esparsos. Inicialmente foi exposto o problema em obter um modelo reduzido para um grande número de variáveis, em seguida foram apresentadas as técnicas, *Best Subset Selection*, *Forward Selection*, *Forward Stagewise Selection* e o *Least Angle Regression*, as quais permitem a escolha de variáveis de forma progressiva, utilizando heurísticas que trazem a minimização do erro quadrático médio fazendo uso de poucas variáveis. Essa característica permite que apenas um subconjunto de variáveis seja utilizada durante a estimativa do modelo e, geralmente, resultando em um menor custo computacional. Foi apresentada também a técnica *Backward Elimination*, uma alternativa às técnicas de seleção progressiva. Neste capítulo também são apresentadas as propostas, a quais fazem uso da regressão esparsa multiresposta e do *matching pursuit* para obtenção de um modelo mais compacto, em termos de vetores suporte, para LSSVMs. As propostas também utilizam o *leave-one-out*, uma técnica de validação cruzada que possui um cálculo simples direto chamado *PRESS Statistic*, permitindo assim um menor custo computacional.

4 Resultados e Discussão

Para que ocorra o treinamento das máquinas de vetores suporte, sua formulação faz uso de um ou mais parâmetros, os quais tem extrema importância no desempenho do modelo estimado e na capacidade de generalização e, portanto, sua escolha deve ser feita com o devido cuidado. Apesar disto, não há nenhuma forma rápida e efetiva de fazer a escolha destes valores, necessitando, em geral, de uma busca exaustiva em um intervalo pré-definido (YANG, 2012). Nesta seção é apresentada a metodologia seguida para a obtenção destes parâmetros. Além disso, após a escolha dos parâmetros e da obtenção do modelo criado, têm-se a necessidade de comparar os modelos de maneira mais justa, dessa forma um método estatístico é utilizado para avaliar o desempenho de todos os modelos. Os resultados obtidos são apresentados na forma de tabelas que indicam o tempo de treinamento e teste, o desempenho e o tamanho do modelo. Todas as simulações foram realizadas no MatlabTM em um computador Intel i3 com 8GB de memória.

Na Subseção 4.1 é abordada a metodologia seguida para a obtenção dos parâmetros dos modelos, na subseção seguinte, Subseção 4.2, é exposto a forma como os modelos foram avaliados e finalmente, nas subseções 4.3 e 4.4 são apresentados os resultados obtidos com as simulações em problemas de classificação e regressão.

4.1 Escolha dos parâmetros do modelo

A fim de avaliar o desempenho das máquinas de vetores suporte e compará-las com a proposta, foram realizadas duas simulações com funções de *kernel* distintas para cada um dos modelos. A primeira faz uso de uma função de *kernel* do tipo linear, $K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}^T \mathbf{x}_i$, e o segundo do tipo RBF, $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma^2})$. Para ambas as abordagens foi necessário a escolha do termo de regularização γ , para o qual foi realizada uma busca exponencial em grade no intervalo $[2^{-5}, 2^{12}]$. Nessa busca, o intervalo de pesquisa foi dividido de maneira exponencial e cada um dos valores foi avaliado. Quando utilizado a função RBF, o valor de abertura σ deve ser estimado e, de forma semelhante ao valor de γ , o intervalo de busca para o parâmetro é $[2^{-3}, 2^1]$. Vale a pena ressaltar que, quando é feita a escolha pela função de *kernel* RBF, a busca em grade é feita em duas dimensões, ocorrendo assim a escolha da melhor combinação de γ e σ . Outro ponto importante a ser lembrado é que a escolha dos intervalos de busca foi feita de forma empírica, mas de maneira semelhante a seguida por YANG Yuling WANG (2012). Cada valor de γ , para problemas lineares, ou combinação de (γ, σ) para problemas não lineares com *kernel* RBF, são escolhidos após a realização da busca com validação cruzada.

No tocante às IP-LSSVMs, há um parâmetro extra para definir, o percentual de

vetores que serão removidos pelo modelo. A fim de realizar uma comparação justa com os demais métodos, foram definidas 3 versões para a IP-LSSVM, onde as três versões diferenciam-se pelo percentual de poda utilizado, a saber: IP-LSSVM_{10%}, IP-LSSVM_{20%} e IP-LSSVM_{30%}, para representar 10%, 20% e 30% de poda, respectivamente, em relação a quantidade total de amostras de treinamento.

Para as P-LSSVMs tem-se três parâmetros extras, percentual de poda por iteração, a tolerância ao erro e o tamanho do conjunto de validação. O percentual de poda escolhido foi de 1%. Caso o erro fosse maior que 2,5% em relação ao melhor desempenho alcançado dentro das iterações, o treinamento estava concluído. O cálculo do desempenho foi realizado sobre um conjunto de validação formado por 10% do conjunto de treinamento, escolhido de forma aleatória em cada evento de treinamento.

4.2 Avaliação do desempenho

O desempenho de um classificador é extremamente dependente do conjunto de treinamento, da natureza dos dados, da quantidade e da dimensionalidade das amostras. Isso pode ser constatado ao se realizar o treinamento com diferentes conjuntos de dados, em cada teste, pode-se observar uma grande diferença no desempenho. Além disso, diferentes classificadores, geralmente, apresentam resultados diferentes para um mesmo conjunto de dados. Isso indica que, avaliar classificadores, bem como compará-los, é uma tarefa essencial para se realizar uma boa escolha de um modelo. Esse mesmo problema se aplica a tarefas de regressão.

Em problemas de classificação e regressão é comum que se busque avaliar o desempenho do modelo, para isso, um teste de hipótese pode ser utilizado a fim de comparar dois modelos diferentes de forma mais justa. Toma-se como hipótese nula H_0 , na qual ambos os resultados obtidos pelos modelos são estatisticamente equivalentes. Uma maneira mais simples de realizar a comparação é utilizando um teste paramétrico, tal como o T-Student (GRIGELIONIS, 2013). Neste teste assume-se que as amostras obtidas são uniformemente distribuídas, assim é possível estimar o valor da média (μ_i) e do desvio-padrão (σ_i) para ambos os modelos. Além disso, sujeito a um grau de significância α , é possível determinar se H_0 é válida ou não. No caso de não ser válida se considera a hipótese alternativa H_1 . O teste estatístico de Wilcoxon também chamado de Mann-Whitney-Wilcoxon *rank sum test*, foi proposto por Wilcoxon (1945) este apresenta uma série de vantagens frente a métodos paramétricos como o de T-Student (BLAIR, 1980).

A utilização de métodos não paramétricos se torna interessante uma vez que não se assume nenhuma distribuição em particular, isto é, uma vez que dados podem assumir qualquer distribuição e variância, e não se pode verificar se estes assumem alguma

distribuição conhecida¹, os métodos não paramétricos permitem que o teste de hipótese seja realizado sem fazer suposições sobre os dados.

4.3 Resultados Obtidos em Problemas de Regressão

Nessa seção são apresentadas as simulações que foram realizadas para avaliar a proposta deste trabalho. Para isso, a OP-LSSVR e a MP-LSSVR são aplicadas a diversos conjunto de dados de natureza artificial e conjuntos de dados oriundos de problemas reais, para problemas de regressão.

4.3.1 Conjunto de dados artificiais

Para avaliar a capacidade de solucionar problemas de regressão, a proposta é aplicada em seis conjuntos de dados artificiais. O primeiro conjunto de dados, chamado Artificial I, é um função quadrática, porém seu intervalo escolhido assemelha-se a uma reta. O segundo conjunto, Artificial II, é uma função que representa um semicírculo definida apenas no intervalo $[-2, 2]$. O terceiro conjunto, Artificial III, é um função seno com amplitude crescente definida em $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Já o quarto conjunto, Artificial IV, é a função tangente hiperbólica. O quinto, Artificial V, é uma função seno, porém definida em um intervalo menor que a Artificial III, $[0, \frac{\pi}{2}]$. Finalmente, o conjunto Artificial VI é uma função formada pela soma de senos, na qual foram utilizados os 5 primeiros termos da série $y_i = \sum_{j=1,3,5,\dots}^{\infty} \frac{\sin(jx_i)}{j}$. Os conjuntos artificiais I, III e V sofreram a adição de ruído gaussiano multiplicativo, enquanto os conjuntos II, IV e VI, sofreram adição de ruído gaussiano aditivo. Na Tabela 2 são descritos os detalhes de cada conjunto de dados.

Tabela 2 – Descrição dos problemas artificiais. $rand_i$ é uma amostra aleatória do ruído.

Nome	Ruído Gaussiano $\rho(\mu, \sigma)$	Função	Domínio
Artificial I	$\rho(0.5, 0.5)$	$y_i = x_i(1.5x_i + 3 + rand_i)$	$[1, 3]$
Artificial II	$\rho(0, 0.8)$	$y_i = 2 - 1 - \frac{x_i^2}{4} + rand_i$	$[-2, 2]$
Artificial III	$\rho(0, 0.5)$	$y_i = i \times (4\sin(x_i) + rand_i)$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
Artificial IV	$\rho(-0.16, 0.08)$	$y_i = \tanh(x_i) + rand_i$	$[-3, 3]$
Artificial V	$\rho(0, 0.5)$	$y_i = i \times (4\sin(x_i) + rand_i)$	$[0, 2\pi]$
Artificial VI	$\rho(0.05, 0.025)$	$y_i = \sum_{j=1,3,5,\dots}^{\infty} \frac{\sin(jx_i)}{j} + rand_i$	$[0, 2\pi]$

A fim de analisar o processo de treinamento, a LSSVR e a OP-LSSVR foram aplicadas a conjuntos de dados artificiais que podem ser visualizados em um plano. Essa característica é importante pois permite que a capacidade de generalização seja avaliada também de forma visual.

No primeiro experimento, a LSSVR e a OP-LSSVR foram treinadas utilizando um *kernel* do tipo RBF. Os hiperparâmetros γ e σ foram obtidos após a realização da busca

¹ Como por exemplo, normal, uniforme, exponencial.

em grade com validação cruzada. Mais detalhes sobre a otimização dos hiperparâmetros podem ser obtidas na subseção 4.1

Nas figuras 8 e 9, são apresentados os resultados obtidos com os conjuntos de dados artificiais. Os pontos em azul são os padrões utilizados durante o treinamento e, portanto, também são os vetores suporte utilizados pela LSSVR. Os pontos em vermelho, em bem menor quantidade, são os vetores suporte escolhidos pela OP-LSSVR.

A curva estimada pela LSSVR é apresentada em verde e em preto a curva obtida pela OP-LSSVR. Pode-se verificar que as curvas obtidas pela OP-LSSVR são semelhantes as obtidas pela LSSVR e, da mesma forma, representam bem o conjunto de treinamento. Além disso, pode-se perceber que os resultados obtidos pela OP-LSSVR são mais suaves do que a da LSSVR, no que diz respeito a curva estimada, figuras 9(c) e 9(e).

A imagem apresentada ao lado direito da curva estimada exibe o erro quadrático médio obtido à medida que a quantidade de vetores suporte são adicionados ao modelo. Pode-se perceber uma rápida redução do erro com a adição de poucos vetores suportes ao modelo, principalmente em modelos mais simples. Em todos os exemplos há pelo menos um ponto em que o erro obtido pela OP-LSSVR é menor que o obtido pela LSSVR, linha vermelha. Além disso, pode-se notar que, a escolha de um outro critério para seleção da quantidade de vetores suporte, como por exemplo um limite máximo de erro quadrático médio em vez do erro quadrático médio mínimo, pode trazer um modelo ainda mais esparsos sem perder a generalização.

4.3.2 Conjunto de dados reais

Os resultados para as simulações com conjuntos de dados reais são apresentados nessa seção. Para isto, dado um conjunto de dados, este foi dividido em outros dois, seguindo uma proporção de dois terços para treinamento e um terço para teste. A seleção de qual padrão pertence a qual conjunto foi feita de forma aleatória em cada um das 30 realizações.

As simulações foram realizadas com onze conjuntos de dados, são estes: Auto price (AUT), Servo (SER), Breast Cancer Wisconsin (BCR), CPU (CPU), Boston (BOS), Stock (STO), Abalone (ABA), Bank (BAN), Ailerons (AIL), Elevators (ELE), and Computer (COM) que são constituintes do repositório UCI Machine Learning. Mais detalhes sobre a quantidade de padrões e dimensionalidade dos dados podem ser observadas na Tabela 3.

Nas tabelas 4, 5, 6 e 7 são apresentados o desempenho, em termos do erro quadrático médio e do desvio padrão para cada uma das técnicas anteriormente apresentadas com funções de *kernel* linear e RBF. As técnicas utilizadas são SVR, LSSVR, P-LSSVR, IP-LSSVR e a proposta. É apresentado também a quantidade média de vetores suporte (#SVs) e o tempo de treinamento e teste.

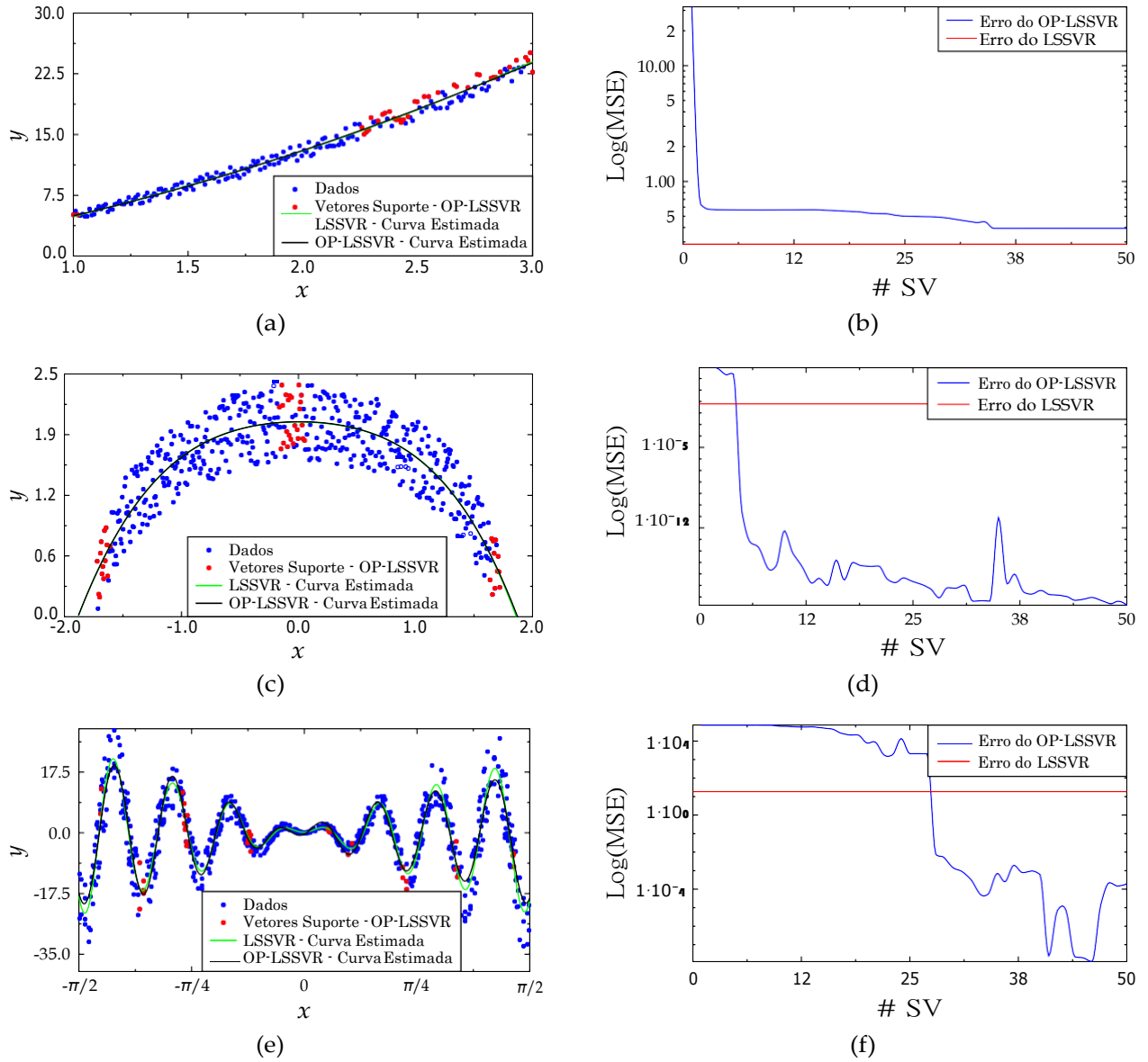


Figura 8 – Resultados das simulações utilizando os 3 primeiros conjuntos artificiais apresentados na Tabela 2. Do lado esquerdo: os pontos azuis são os dados de treinamento. Em vermelho os vetores suporte utilizados pela OP-LSSVR para estimar a curva em verde. Em preto a curva estimada pela LSSVR. Do lado direito: erro obtido em função da quantidade de vetores adicionados para estimar a curva à esquerda

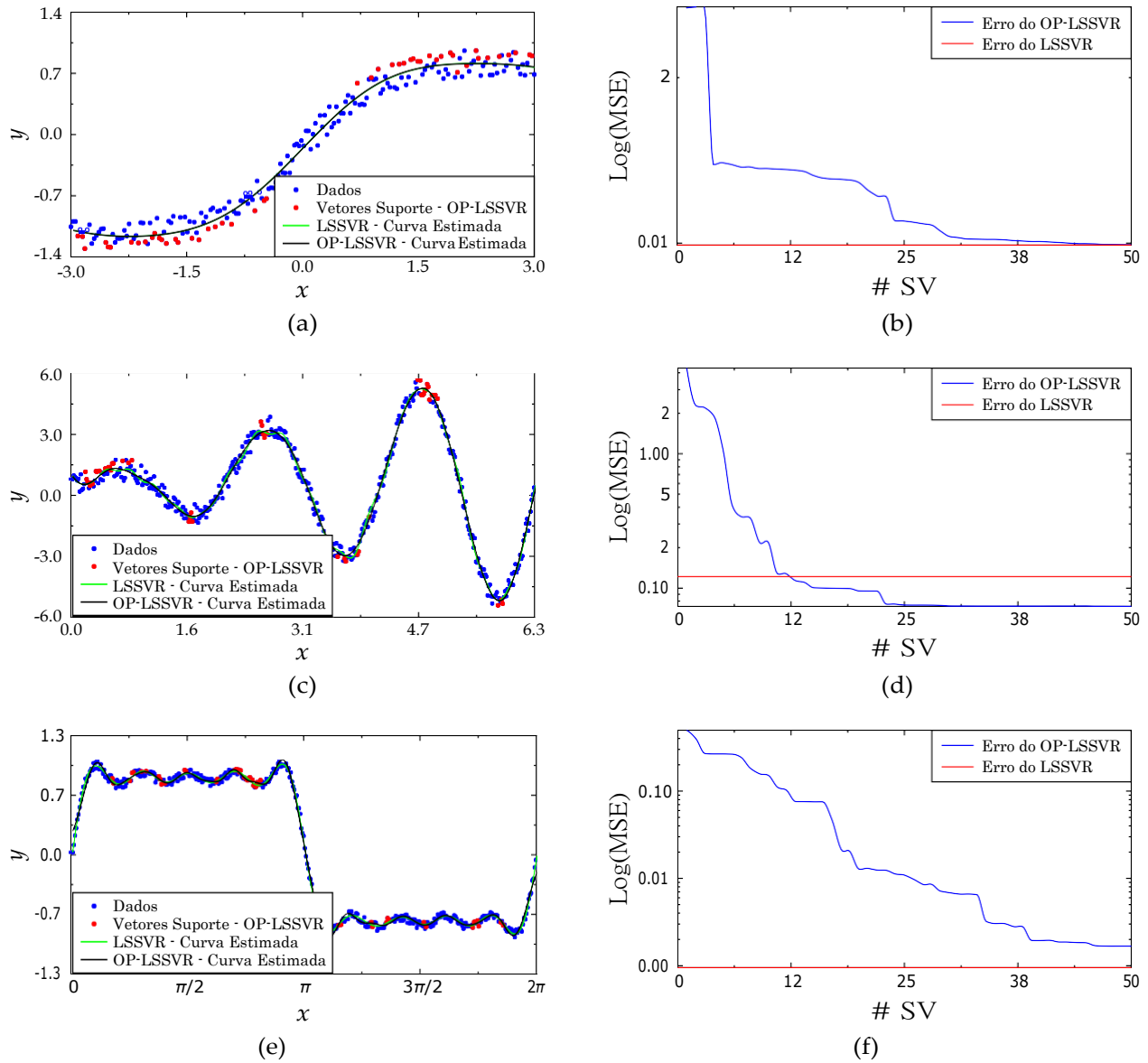


Figura 9 – Resultados das simulações utilizando os 3 últimos conjuntos artificiais apresentados na Tabela 2. Do lado esquerdo: os pontos azuis são os dados de treinamento. Em vermelho os vetores suporte utilizados pela OP-LSSVR para estimar a curva em verde. Em preto a curva estimada pela LSSVR. Do lado direito: erro obtido em função da quantidade de vetores adicionados para estimar a curva à esquerda

Tabela 3 – Descrição dos conjuntos de dados reais utilizados para regressão.

Nome do Conj.	#Amostras	#Treinamento	#Teste	#Variaveis
Auto P.	159	104	55	15
Servo	167	110	57	4
CPU	209	137	72	6
Breast C.	194	128	66	32
Boston	506	333	173	13
Stock	950	627	323	9
Abalone	4177	2756	1421	8
Bank	4499	2969	1530	8
Ailerons	7129	4705	2424	5
Computer	8192	5406	2786	12
Elevators	9517	6281	3236	6

Uma vez que a OP-LSSVR e a MP-LSSVR fazem uso de apenas uma pequena parcela de vetores suporte, foi escolhido inicialmente para o processo de otimização do *Leave-one-out* apenas os 100 melhores, selecionados pelo MRSR e MP. Essa metodologia só não foi seguida para as bases da BAN e COM quando avaliadas com o *kernel* RBF, onde neste caso foram utilizados 200 vetores suporte. Isso se fez necessário pois o desempenho da OP-LSSVR e da MP-LSSVR não apresentaram-se adequados.

O teste estatístico de Wilcoxon foi utilizado para comparar a proposta (OP-LSSVR). O Wilcoxon *Rank sum test* (WILCOXON, 1945), é um teste de hipótese não paramétrico que verifica se dois conjuntos de dados de diferentes populações são equivalentes. A metodologia proposta neste trabalho faz uso dos resultados obtidos, ou seja, do erro quadrático médio como amostra para a verificação do teste. Os resultados do teste são apresentados nas tabelas 4, 5, 6 e 7 considerando um nível de significância de 5%. Em cada célula da tabela é verificado se o resultado obtido pela OP-LSSVR é equivalente ao apresentado por essa célula. As células que apresentam um • apontam um resultado inferior por parte da OP-LSSVR. De uma forma geral, pode-se concluir que o resultado obtido pela OP-LSSVR, em termos de erro quadrático médio, é equivalente ou superior a todas as outras técnicas, a saber: SVR, LSSVR, P-LSSVR, IP-LSSVR e MP-LSSVR. Essa análise só não se confirma em 5 resultados obtidos, 3 destes na abordagem linear da OP-LSSVR e outros dois na abordagem com *kernel* RBF.

Tabela 4 – Resultado *kernel* Linear Parte 1. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando *kernel* linear.

O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.

Regressor		AUT	SER	BCR	CPU	BOS
SVM	MSE	4,2e+07 ± 1,0e+07	2,4e+00 ± 6,3e-01	1,1e+03 ± 1,3e+02	2,8e+04 ± 1,2e+04	3,4e+01 ± 6,7e+00
	#SVs	94,0 ± 0,0	96,5 ± 1,4	116,0 ± 0,2	124,1 ± 0,3	300,3 ± 0,9
	Tempo	1,5e-03 ± 1,6e-03	7,5e-04 ± 3,4e-04	1,5e-03 ± 1,9e-04	1,1e-03 ± 6,0e-04	5,9e-03 ± 1,9e-03
LSSVR	MSE	7,0e+06 ± 2,0e+06	1,4e+00 ± 2,7e-01	1,1e+03 ± 1,6e+02	5,3e+03 ± 2,9e+03	2,4e+01 ± 4,1e+00
	#SVs	104,0 ± 0,0	110,0 ± 0,0	128,0 ± 0,0	137,0 ± 0,0	333,0 ± 0,0
	Tempo	1,3e-03 ± 1,7e-03	1,1e-03 ± 1,0e-04	1,3e-03 ± 8,5e-05	1,4e-03 ± 5,4e-05	8,6e-03 ± 1,1e-03
IP-LSSVR _{10%}	MSE	6,7e+07 ± 7,8e+07	2,0e+00 ± 1,1e+00	1,9e+04 ± 1,4e+04	2,3e+04 ± 1,1e+04	3,2e+02 ± 2,1e+02
	#SVs	94,0 ± 0,0	99,0 ± 0,0	115,0 ± 0,0	123,0 ± 0,0	300,0 ± 0,0
	Tempo	9,1e-04 ± 4,5e-05	9,5e-04 ± 3,4e-05	1,3e-03 ± 4,2e-04	1,3e-03 ± 5,0e-05	7,5e-03 ± 5,1e-04
IP-LSSVR _{20%}	MSE	3,1e+08 ± 5,8e+08	2,6e+00 ± 7,1e-01	1,4e+04 ± 1,1e+04	2,3e+04 ± 1,4e+04	1,5e+03 ± 5,8e+03
	#SVs	83,0 ± 0,0	88,0 ± 0,0	102,0 ± 0,0	110,0 ± 0,0	266,0 ± 0,0
	Tempo	8,8e-04 ± 4,0e-05	9,4e-04 ± 2,9e-05	1,2e-03 ± 6,1e-05	1,2e-03 ± 4,2e-05	6,7e-03 ± 7,5e-04
IP-LSSVR _{30%}	MSE	1,5e+08 ± 2,1e+08	3,0e+00 ± 9,4e-01	1,5e+04 ± 9,7e+03	2,1e+04 ± 1,0e+04	1,2e+03 ± 2,1e+03
	#SVs	73,0 ± 0,0	77,0 ± 0,0	90,0 ± 0,0	96,0 ± 0,0	233,0 ± 0,0
	Tempo	8,6e-04 ± 4,5e-05	9,3e-04 ± 4,7e-05	1,2e-03 ± 6,0e-05	1,2e-03 ± 5,0e-05	6,2e-03 ± 2,5e-04
P-LSSVR	MSE	1,1e+07 ± 4,6e+06	2,1e+00 ± 7,2e-01	1,5e+03 ± 5,2e+02	8,1e+03 ± 4,6e+03	4,0e+01 ± 1,7e+01
	#SVs	76,7 ± 0,0	68,6 ± 0,0	80,7 ± 0,0	95,6 ± 0,0	241,0 ± 0,0
	Tempo	1,1e-02 ± 7,4e-03	1,4e-02 ± 5,5e-03	1,1e-02 ± 4,3e-03	8,3e-03 ± 7,0e-03	6,2e-02 ± 2,4e-02
OP-LSSVR	MSE	7,9e+06 ± 1,7e+06	1,3e+00 ± 2,1e-01	1,1e+03 ± 1,7e+02	6,4e+03 ± 3,4e+03	2,4e+01 ± 4,6e+00
	#SVs	50,5 ± 23,8	28,4 ± 18,8	5,5 ± 10,9	29,0 ± 19,3	43,4 ± 7,3
	Tempo	6,8e-02 ± 1,5e-02	3,1e-02 ± 1,2e-02	3,6e-02 ± 4,3e-03	3,8e-02 ± 3,7e-03	8,2e-02 ± 1,7e-02
MP-LSSVR	MSE	7,9e+06 ± 2,4e+06	1,4e+00 ± 3,0e-01	1,1e+03 ± 1,6e+02	• 4,7e+03 ± 2,9e+03	2,6e+01 ± 5,4e+00
	#SVs	34,8 ± 24,8	10,3 ± 6,4	22,0 ± 26,3	14,4 ± 11,9	17,7 ± 17,2
	Tempo	5,0e-02 ± 8,7e-03	1,0e-01 ± 3,4e-02	9,2e-02 ± 2,6e-02	1,4e-01 ± 3,9e-02	1,4e-01 ± 3,9e-02

Tabela 5 – Resultado *kernel* Linear Parte 2. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando *kernel* linear. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.

Regressor		STO	ABA	AIL	ELE	BAN	COM
SVM	MSE	7,0e+00 ± 4,7e-01	5,5e+00 ± 4,4e-01	1,0e-07 ± 3,7e-08	4,7e-06 ± 1,8e-06	1,6e-03 ± 1,1e-04	1,4e+02 ± 1,2e+01
	#SVs	563,5 ± 1,2	2468,6 ± 2,8	2,0 ± 0,5	1,9 ± 0,4	2033,9 ± 15,9	4850,0 ± 4,2
	Tempo	1,8e-02 ± 8,5e-03	3,7e-01 ± 2,0e-01	1,2e-03 ± 6,9e-04	1,8e-03 ± 4,5e-04	6,4e-01 ± 3,3e-01	1,6e+00 ± 7,9e-01
LSSVR	MSE	5,5e+00 ± 3,7e-01	5,0e+00 ± 2,7e-01	3,0e-08 ± 1,6e-09	2,1e-06 ± 4,8e-08	1,5e-03 ± 1,1e-04	9,9e+01 ± 7,3e+00
	#SVs	627,0 ± 0,0	2756,0 ± 0,0	4705,0 ± 0,0	6281,0 ± 0,0	2969,0 ± 0,0	5406,0 ± 0,0
	Tempo	4,9e-02 ± 4,3e-03	2,5e+00 ± 1,7e-01	1,1e+01 ± 8,1e-01	2,4e+01 ± 1,1e+00	2,9e+00 ± 1,6e-01	1,5e+01 ± 6,9e-01
IP-LSSVR _{10%}	MSE	2,9e+03 ± 5,2e+03	9,5e+03 ± 2,9e+03	4,6e-06 ± 1,3e-06	1,8e-04 ± 6,4e-05	3,4e-01 ± 9,7e-02	7,7e+05 ± 1,5e+05
	#SVs	564,0 ± 0,0	2480,0 ± 0,0	4234,0 ± 0,0	5653,0 ± 0,0	2672,0 ± 0,0	4865,0 ± 0,0
	Tempo	4,5e-02 ± 4,6e-03	2,1e+00 ± 9,4e-02	8,9e+00 ± 3,3e-01	2,0e+01 ± 5,1e-01	2,5e+00 ± 5,0e-02	1,3e+01 ± 3,2e-01
IP-LSSVR _{20%}	MSE	4,7e+03 ± 7,0e+03	8,8e+03 ± 4,5e+03	5,7e-06 ± 1,4e-06	1,2e-04 ± 4,4e-05	2,5e-01 ± 6,4e-02	4,8e+05 ± 1,5e+05
	#SVs	502,0 ± 0,0	2205,0 ± 0,0	3764,0 ± 0,0	5025,0 ± 0,0	2375,0 ± 0,0	4325,0 ± 0,0
	Tempo	3,7e-02 ± 2,6e-03	1,7e+00 ± 2,9e-02	7,6e+00 ± 2,7e-01	1,7e+01 ± 5,7e-01	2,2e+00 ± 9,4e-02	1,1e+01 ± 4,2e-01
IP-LSSVR _{30%}	MSE	6,0e+03 ± 6,8e+03	4,5e+03 ± 3,5e+03	5,1e-06 ± 1,4e-06	7,5e-05 ± 3,3e-05	1,8e-01 ± 5,4e-02	3,5e+05 ± 7,9e+04
	#SVs	439,0 ± 0,0	1929,0 ± 0,0	3293,0 ± 0,0	4397,0 ± 0,0	2078,0 ± 0,0	3784,0 ± 0,0
	Tempo	3,5e-02 ± 4,8e-03	1,6e+00 ± 6,7e-02	6,5e+00 ± 2,1e-01	1,5e+01 ± 4,9e-01	1,9e+00 ± 8,2e-02	9,7e+00 ± 3,8e-01
P-LSSVR	MSE	8,2e+00 ± 2,3e+00	7,8e+00 ± 9,5e-01	4,4e-08 ± 3,0e-09	3,1e-06 ± 1,2e-07	2,3e-03 ± 3,1e-04	1,4e+02 ± 3,1e+01
	#SVs	507,2 ± 0,0	1842,7 ± 0,0	2572,3 ± 0,0	3089,9 ± 0,0	2016,9 ± 0,0	4555,7 ± 0,0
	Tempo	2,2e-01 ± 7,9e-02	1,8e+01 ± 6,9e+00	8,9e+01 ± 1,9e+01	2,0e+02 ± 3,7e+01	2,0e+01 ± 5,9e+00	3,5e+01 ± 3,6e+01
OP-LSSVR	MSE	5,5e+00 ± 3,8e-01	5,0e+00 ± 2,5e-01	3,0e-08 ± 1,5e-09	2,1e-06 ± 5,2e-08	1,6e-03 ± 1,2e-04	1,0e+02 ± 7,6e+00
	#SVs	40,8 ± 7,7	35,7 ± 12,3	22,7 ± 18,5	20,0 ± 17,3	34,7 ± 14,6	39,9 ± 9,6
	Tempo	3,5e-01 ± 4,2e-02	9,4e+00 ± 1,1e+00	3,8e+01 ± 3,5e+00	6,4e+01 ± 1,2e+01	5,9e+00 ± 2,7e-01	2,0e+01 ± 2,4e-01
MP-LSSVR	MSE	6,4e+00 ± 2,1e+00	4,8e+00 ± 2,4e-01	3,0e-08 ± 9,3e-10	2,1e-06 ± 5,2e-08	1,6e-03 ± 9,7e-05	9, 6e + 01 ± 7,2e+00
	#SVs	22,8 ± 16,2	27,4 ± 19,6	20,3 ± 21,2	18,9 ± 19,1	59,9 ± 24,7	57,8 ± 26,3
	Tempo	3,6e-01 ± 8,1e-02	5,2e+00 ± 1,0e+00	1,3e+01 ± 3,8e+00	2,7e+01 ± 1,0e+01	5,2e+00 ± 2,4e+00	2,0e+01 ± 8,7e+00

Tabela 6 – Resultado *kernel* RBF Parte 1. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando *kernel* RBF. O resultados com um ponto preto na frete, (●), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.

Regressor		AUT	SER	BCR	CPU	BOS
SVM	MSE	2,4e+07 ± 3,4e+06	4,9e-01 ± 1,5e-01	1,2e+03 ± 2,3e+02	9,0e+03 ± 4,1e+03	1,3e+01 ± 4,8e+00
	#SVs	94,0 ± 0,0	71,8 ± 6,0	116,0 ± 0,0	123,0 ± 1,2	290,0 ± 2,6
	Tempo	1,4e-03 ± 7,6e-04	1,9e-03 ± 1,0e-03	1,9e-03 ± 6,4e-04	4,6e-03 ± 4,3e-03	1,4e-02 ± 9,2e-04
LSSVR	MSE	7,3e+06 ± 3,7e+06	5,0e-01 ± 2,5e-01	1,1e+03 ± 1,8e+02	7,3e+03 ± 6,5e+03	1,2e+01 ± 4,1e+00
	#SVs	104,0 ± 0,0	110,0 ± 0,0	128,0 ± 0,0	137,0 ± 0,0	333,0 ± 0,0
	Tempo	1,8e-03 ± 8,9e-04	1,7e-03 ± 2,6e-04	3,3e-03 ± 2,5e-03	2,4e-03 ± 1,1e-03	1,5e-02 ± 3,8e-03
IP-LSSVR _{10%}	MSE	1,2e+07 ± 5,4e+06	1,2e+00 ± 5,7e-01	1,6e+03 ± 2,0e+03	1,4e+04 ± 7,1e+03	7,7e+01 ± 8,0e+01
	#SVs	94,0 ± 0,0	99,0 ± 0,0	115,0 ± 0,0	123,0 ± 0,0	300,0 ± 0,0
	Tempo	1,5e-03 ± 3,9e-04	1,4e-03 ± 6,3e-05	2,1e-03 ± 2,8e-04	1,9e-03 ± 2,2e-04	1,0e-02 ± 5,3e-04
IP-LSSVR _{20%}	MSE	1,4e+07 ± 7,2e+06	1,8e+00 ± 1,3e+00	2,0e+03 ± 2,3e+03	1,4e+04 ± 9,1e+03	6,5e+01 ± 2,5e+01
	#SVs	83,0 ± 0,0	88,0 ± 0,0	102,0 ± 0,0	110,0 ± 0,0	266,0 ± 0,0
	Tempo	1,7e-03 ± 2,4e-04	1,7e-03 ± 7,5e-04	2,5e-03 ± 6,8e-04	2,1e-03 ± 3,7e-04	1,1e-02 ± 1,8e-03
IP-LSSVR _{30%}	MSE	1,6e+07 ± 1,1e+07	2,0e+00 ± 1,5e+00	2,2e+03 ± 2,5e+03	1,2e+04 ± 9,0e+03	8,7e+01 ± 1,4e+02
	#SVs	73,0 ± 0,0	77,0 ± 0,0	90,0 ± 0,0	96,0 ± 0,0	233,0 ± 0,0
	Tempo	1,4e-03 ± 3,0e-04	1,3e-03 ± 6,7e-05	2,2e-03 ± 3,5e-04	1,8e-03 ± 4,6e-04	8,9e-03 ± 5,3e-04
P-LSSVR	MSE	1,5e+07 ± 6,9e+06	5,8e-01 ± 2,5e-01	1,3e+03 ± 2,4e+02	1,1e+04 ± 8,6e+03	2,3e+01 ± 5,6e+00
	#SVs	84,6 ± 0,0	97,4 ± 0,0	99,7 ± 0,0	113,3 ± 0,0	291,4 ± 0,0
	Tempo	1,8e-02 ± 1,4e-02	6,2e-03 ± 3,3e-03	2,1e-02 ± 1,4e-02	1,3e-02 ± 1,5e-02	4,7e-02 ± 3,6e-02
OP-LSSVR	MSE	7,4e+06 ± 2,5e+06	4,3e-01 ± 1,2e-01	1,1e+03 ± 1,4e+02	5,7e+03 ± 5,3e+03	1,2e+01 ± 4,1e+00
	#SVs	28,8 ± 16,7	45,7 ± 5,2	9,5 ± 4,6	31,1 ± 17,0	84,0 ± 16,7
	Tempo	3,0e-02 ± 3,0e-03	3,8e-02 ± 7,9e-03	3,4e-02 ± 2,7e-03	5,2e-02 ± 5,5e-03	2,7e-01 ± 4,5e-02
MP-LSSVR	MSE	1,1e+07 ± 3,9e+06	8,7e-01 ± 2,8e-01	1,1e+03 ± 1,3e+02	1,1e+04 ± 6,9e+03	2,7e+01 ± 9,0e+00
	#SVs	6,3 ± 4,9	15,0 ± 7,2	12,8 ± 14,8	6,6 ± 3,8	24,7 ± 18,9
	Tempo	6,2e-02 ± 1,4e-02	5,5e-02 ± 2,3e-02	8,4e-02 ± 5,9e-02	1,2e-01 ± 5,7e-02	8,8e-02 ± 2,6e-02

Tabela 7 – Resultado *kernel* RBF Parte 2. Erro quadrático médio e desvio padrão após 30 execuções independentes utilizando *kernel* RBF. O resultados com um ponto preto na frete, (●), apontam um desempenho pior, em termos do erro quadrático médio, para a OP-LSSVR em relação a técnica indicada. Para todo os outros casos a OP-LSSVR é melhor ou equivalente.

Regressor		STO	ABA	AIL	ELE	BAN	COM
SVM	MSE	6,3e-01 ± 1,2e-01	4,6e+00 ± 1,7e-01	5,3e-08 ± 6,0e-09	2,1e-06 ± 2,8e-08	1,6e-03 ± 1,7e-04	3,3e+01 ± 2,3e+00
	#SVs	491,3 ± 5,7	2338,8 ± 6,6	689,5 ± 503,2	2463,5 ± 108,2	33,0 ± 3,6	4654,8 ± 13,3
	Tempo	5,3e-02 ± 1,6e-02	3,2e-01 ± 2,1e-02	3,5e+00 ± 6,9e+00	3,4e+01 ± 3,1e+01	6,1e-02 ± 3,9e-03	8,3e+00 ± 3,5e+00
LSSVR	MSE	5,0e-01 ± 5,9e-02	4,5e+00 ± 1,9e-01	2,8e-08 ± 1,1e-09	2,0e-06 ± 5,8e-08	● 9,5e-04 ± 6,0e-05	● 9,7e+00 ± 6,0e-01
	#SVs	627,0 ± 0,0	2756,0 ± 0,0	4705,0 ± 0,0	6281,0 ± 0,0	2969,0 ± 0,0	5406,0 ± 0,0
	Tempo	5,9e-02 ± 6,8e-03	2,1e+00 ± 1,6e-01	1,0e+01 ± 3,1e-01	2,4e+01 ± 7,1e-01	3,1e+00 ± 1,9e-01	1,6e+01 ± 6,1e-01
IP-LSSVR _{10%}	MSE	1,4e+01 ± 2,6e+01	6,7e+00 ± 6,4e-01	5,4e-08 ± 2,9e-09	3,4e-06 ± 1,6e-07	2,9e-03 ± 2,7e-04	1,3e+03 ± 4,5e+03
	#SVs	564,0 ± 0,0	2480,0 ± 0,0	4234,0 ± 0,0	5653,0 ± 0,0	2672,0 ± 0,0	4865,0 ± 0,0
	Tempo	5,5e-02 ± 4,0e-03	2,2e+00 ± 1,1e-01	1,0e+01 ± 4,4e+00	1,7e+01 ± 1,4e+00	2,3e+00 ± 1,4e-01	4,3e+01 ± 1,6e+01
IP-LSSVR _{20%}	MSE	1,7e+01 ± 2,3e+01	6,9e+00 ± 9,3e-01	5,7e-08 ± 3,2e-09	3,8e-06 ± 1,5e-07	5,1e-03 ± 1,2e-03	1,4e+04 ± 6,7e+04
	#SVs	502,0 ± 0,0	2205,0 ± 0,0	3764,0 ± 0,0	5025,0 ± 0,0	2375,0 ± 0,0	4325,0 ± 0,0
	Tempo	5,0e-02 ± 6,6e-03	1,7e+00 ± 1,2e-01	8,3e+00 ± 1,9e+00	1,7e+01 ± 5,8e-01	2,3e+00 ± 1,1e-01	4,7e+01 ± 2,3e+01
IP-LSSVR _{30%}	MSE	2,1e+02 ± 7,1e+02	6,9e+00 ± 1,3e+00	5,9e-08 ± 3,4e-09	4,2e-06 ± 8,0e-07	5,6e-03 ± 1,1e-03	2,1e+03 ± 3,1e+03
	#SVs	439,0 ± 0,0	1929,0 ± 0,0	3293,0 ± 0,0	4397,0 ± 0,0	2078,0 ± 0,0	3784,0 ± 0,0
	Tempo	4,4e-02 ± 2,5e-03	1,5e+00 ± 1,1e-01	5,9e+00 ± 5,4e-01	1,7e+01 ± 5,7e+00	2,5e+00 ± 1,1e+00	3,6e+01 ± 2,4e+01
P-LSSVR	MSE	5,4e-01 ± 6,9e-02	4,6e+00 ± 2,3e-01	3,0e-08 ± 2,5e-09	2,1e-06 ± 7,9e-08	2,0e-03 ± 2,1e-04	3,9e+01 ± 4,4e+00
	#SVs	559,0 ± 0,0	2443,5 ± 0,0	4190,6 ± 0,0	5609,3 ± 0,0	2645,1 ± 0,0	4679,8 ± 0,0
	Tempo	8,5e-02 ± 3,5e-02	2,0e+00 ± 1,1e+00	6,0e+00 ± 3,1e+00	1,1e+01 ± 5,5e+00	2,4e+00 ± 7,9e-01	1,5e+01 ± 9,9e+00
OP-LSSVR	MSE	5,9e-01 ± 8,5e-02	4,5e+00 ± 2,2e-01	2,7e-08 ± 1,2e-09	2,1e-06 ± 5,0e-08	1,0e-03 ± 2,9e-05	2,1e+01 ± 8,4e+00
	#SVs	198,7 ± 2,3	47,8 ± 3,8	48,1 ± 2,6	45,9 ± 4,4	189,3 ± 10,1	137,3 ± 31,1
	Tempo	1,7e+00 ± 3,0e-01	8,0e+00 ± 7,8e-01	2,3e+01 ± 2,3e+00	5,0e+01 ± 1,3e+01	2,6e+01 ± 7,1e+00	8,2e+01 ± 8,6e+00
MP-LSSVR	MSE	3,6e+00 ± 2,0e+00	4,9e+00 ± 2,7e-01	3,2e-08 ± 2,0e-09	2,1e-06 ± 6,1e-08	2,7e-03 ± 7,9e-04	7,7e+01 ± 2,3e+01
	#SVs	43,5 ± 26,7	23,3 ± 11,9	22,4 ± 16,8	35,3 ± 22,1	29,7 ± 24,4	37,1 ± 25,3
	Tempo	1,9e-01 ± 5,6e-02	3,6e+00 ± 9,4e-01	9,9e+00 ± 2,5e+00	1,4e+01 ± 3,4e+00	3,5e+00 ± 1,7e+00	1,2e+01 ± 3,8e+00

Tabela 8 – Percentual de redução entre o regressor apresentado em cada linha e a LSSVR com *kernel* linear.

Regressor	AUT	SER	BCR	CPU	BOS	STO	ABA	AIL	ELE	BAN	COM
SVM	9,62	12,27	9,40	9,39	9,82	10,13	10,43	99,96	99,97	31,50	10,29
LSSVR	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
IP-LSSVR _{10%}	9,62	10,00	10,16	10,22	9,91	10,05	10,01	10,01	10,00	10,00	10,01
IP-LSSVR _{20%}	20,19	20,00	20,31	19,71	20,12	19,94	19,99	20,00	20,00	20,01	20,00
IP-LSSVR _{30%}	29,81	30,00	29,69	29,93	30,03	29,98	30,01	30,01	30,00	30,01	30,00
P-LSSVR	26,28	37,64	36,98	30,22	27,63	19,11	33,14	45,33	50,81	32,07	15,73
OP-LSSVR	51,47	74,15	95,70	78,86	86,98	93,50	98,71	99,52	99,68	98,83	99,26
MP-LSSVR	66,51	90,67	82,84	89,51	94,67	96,36	99,01	99,57	99,70	97,98	98,93

Tabela 9 – Percentual de redução entre o regressor apresentado em cada linha e a LSSVR com *kernel* RBF.

Regressor	AUT	SER	BCR	CPU	BOS	STO	ABA	AIL	ELE	BAN	COM
SVM	9,62	34,77	9,38	10,22	12,91	21,65	15,14	85,35	60,78	98,89	13,90
LSSVR	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
IP-LSSVR _{10%}	9,62	10,00	10,16	10,22	9,91	10,05	10,01	10,01	10,00	10,00	10,01
IP-LSSVR _{20%}	20,19	20,00	20,31	19,71	20,12	19,94	19,99	20,00	20,00	20,01	20,00
IP-LSSVR _{30%}	29,81	30,00	29,69	29,93	30,03	29,98	30,01	30,01	30,00	30,01	30,00
P-LSSVR	18,62	11,42	22,08	17,32	12,49	10,85	11,34	10,93	10,69	10,91	13,43
OP-LSSVR	72,34	58,42	92,60	77,32	86,35	92,08	98,27	98,98	99,27	98,36	99,14
MP-LSSVR	93,94	86,39	89,97	95,18	92,59	93,07	99,15	99,52	99,44	99,00	99,31

Na Tabelas 8 e 9 são apresentados o percentual de redução em relação a quantidade de vetores suporte utilizados pela LSSVR. A OP-LSSVR obteve uma redução entre 51,47% (AUT) e 99,26% (COM) para *kernel* lineares. Em relação a *kernel* RBF, a OP-LSSVR obteve um redução entre 58,42% (SER) e 99,27% (ELE). Em termos de esparsidade, a MP-LSSVR provê soluções bem mais esparsas que as outras técnicas tanto para a versão linear quanto RBF do *kernel*. Essa afirmação só não é válida para os conjuntos de dados BCR, BAN e COM para o caso linear e BCR para o caso RBF.

Apesar da MP-LSSVR obter altas taxas de esparsidade, inclusive em relação a OP-LSSVR, o MSE obtido pela MP-LSSVR é, na grande maioria dos casos, maior. De fato, a OP-LSSVR alcança uma maior taxa de esparsidade com um modelo equivalente exceto para (ABA, CPU e BOS) no caso linear e para (BAN e COM) no caso RBF.

Em relação ao tempo necessário para o treinamento e teste, pode-se notar que a LSSVR e a OP-LSSVR são as metodologias que mais fazem uso deste recurso. Na Tabela 6, para um conjunto de dados pequeno (AUT, SER, BCR e CPU), pode-se notar que a OP-LSSVR leva mais tempo para treinar do que a LSSVR, quase uma ordem de magnitude. De maneira similar, para conjunto de dados grandes (ABA, AIL e ELE), Tabela 7, o tempo de treinamento é aproximadamente 4 vezes maior para o ABA, 2 vezes para o AIL e 2 vezes para o ELE.

Uma análise semelhante pode ser realizada para o tempo de treinamento da OP-LSSVR com *kernel* linear. Veja os resultados na Tabela 4. Como pode-se perceber, para um conjunto de dados pequeno, (AUT, SER, BCR), o tempo de treinamento em relação a LSSVR é cerca de uma ordem de magnitude maior. Contudo, para conjuntos de dados

maiores, Tabela 5, o tempo de treinamento é de 5, 3 e 2 vezes maiores que a LSSVR para o ABA, AIL e ELE respectivamente. Esses resultados indicam que, quanto maior o conjunto de dados menor a diferença de tempo de treinamento entre a OP-LSSVR e a LSSVR. Para a MP-LSSVR pode-se constatar que essa técnica possui um tempo de treinamento menor que as outras metodologia, como esperado, uma vez que o MP possui uma heurística mais simples que o MRSR.

4.4 Resultados Obtidos em Problemas de Classificação

Nesta seção são apresentadas as simulações que foram realizadas para avaliar a proposta deste trabalho quando aplicadas a problemas de classificação. Inicialmente, da mesma forma que a metodologia utilizada em problemas de regressão, a OP-LSSVC foi aplicada à conjuntos de dados de natureza artificial e conjuntos de dados reais.

4.4.1 Conjunto de dados artificiais

Para mostrar a capacidade de solucionar problemas de classificação, a proposta é aplicada em quatro conjuntos de dados de duas dimensões, permitindo assim a fácil visualização da região de decisão.

O primeiro conjunto de dados avaliado é o da Íris, neste conjuntos de dados, 3 espécies de flores são apresentadas e devem ser classificadas de acordo com as características, largura e comprimento da sépala e largura e comprimento da pétala. Uma vez que para a visualização é necessário a escolha de somente duas dimensões, a largura e comprimento da sépala foram utilizados. Além disso é realizada a classificação apenas entre a espécie setosa e outras. É importante ressaltar que a escolha desta duas características torna o problema linearmente classificável.

O segundo conjunto de dados foi criado artificialmente e é caracterizado como duas classes onde cada classe foi criada em torno do ponto (0,0) para a classe 1 e em torno do ponto (1,1) para a classe dois. Este problema também é linearmente separável.

O terceiro conjunto de dados é chamado de problema da banana ou ainda duas luas. Duas classes em formato de lua são formadas e dispostas de forma não linearmente separável.

O quarto e último conjunto, chamado de espiral, é composto por duas classes e os padrões que forma cada classe possuem o formato de espiral e, além disso, estão dispostas uma dentro da outra.

A fim de fazer uma análise qualitativa do processo de treinamento, a OP-LSSVC e a LSSVC foram submetidas aos conjuntos de dados mencionados anteriormente. Esses dados, em sua totalidade são conjuntos bidimensionais, e portanto, podem ser visualizados em

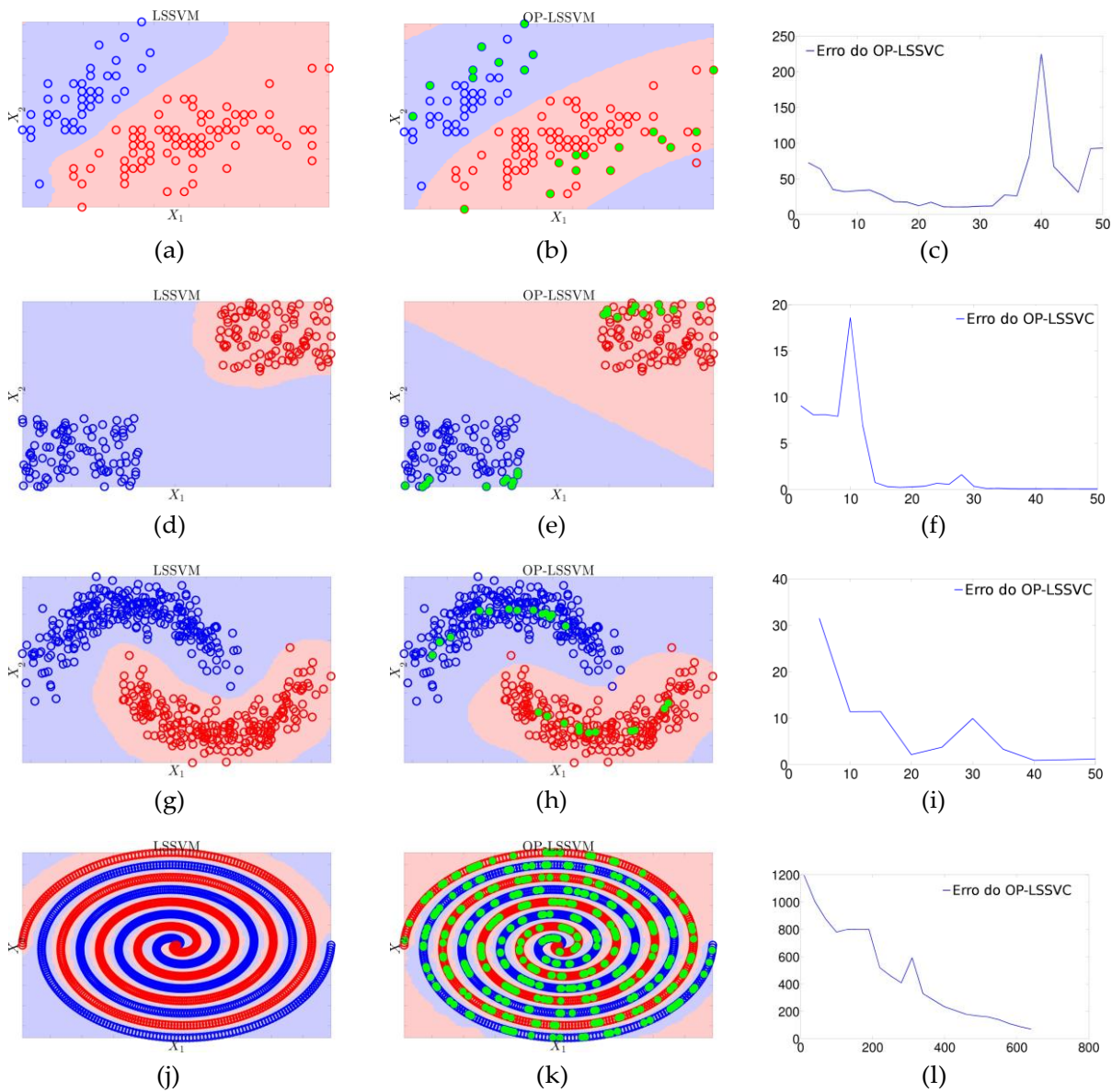


Figura 10 – Resultados das simulações utilizando os 4 conjuntos artificiais de classificação. Do lado esquerdo: os pontos azuis e vermelho são os dados de treinamento. Em verde, ao centro, os vetores suporte utilizados pela OP-LSSVC para estimar a região de decisão. Do lado direito: erro obtido em função da quantidade de vetores adicionados para estimar a função discriminante.

um plano. Essa característica é importante pois permite que a capacidade de generalização seja avaliada também de forma visual.

No primeiro experimento, a LSSVC e a OP-LSSVC foram treinadas utilizando um *kernel* do tipo RBF. Os parâmetros γ e σ foram obtidos após a realização da busca em grade com validação cruzada. Mais detalhes sobre a otimização dos parâmetros podem ser obtidas na subseção 4.1

Na Figura 10 são apresentados os resultados obtidos para os conjuntos de dados

Tabela 10 – Descrição dos conjuntos de dados reais utilizados para classificação.

Nome do Conj.	#Amostras	#Treino	#Teste	#Variáveis
Iris	150	99	51	4
Haberman	306	201	105	3
Vertebral C.	310	204	106	6
Dermatology	366	241	125	34
Breast C.	699	461	238	9
Diabetes	768	506	262	8
Ripley	1250	825	425	2
Spiral	2000	1320	680	2
Banana	5300	3498	1802	2

artificiais. Nas figuras a esquerda são apresentadas as regiões de decisão obtidas com a LSSVC, ao centro, as regiões da OP-LSSVC e a direita, o erro *leave-one-out* durante o processo de seleção dos vetores suporte para a OP-LSSVC. Em ambas as figuras, da esquerda e da direita, pontos em azul e vermelho representam os padrões das classes. Os pontos em verde, apresentados nas figuras ao centro, são os vetores suporte escolhidos para a OP-LSSVC. Uma vez que a LSSVC faz uso de todos os dados de treinamento como vetores suporte, estes não são destacados na figura à direita.

Pode-se verificar que, de uma forma geral a quantidade de vetores suporte utilizado pela OP-LSSVC é significativamente menor, além disso as fronteiras das regiões de decisão são mais suaves, isso pode ser verificado em qualquer uma das figuras ao centro. Um outro ponto que deve ser levantado é que a quantidade de vetores suporte utilizados pela OP-LSSVC pode ser ainda menor. Nota-se nos gráficos a direita o erro em função da quantidade de vetores adicionados, indicando que uma quantidade de vetores suporte menor pode fornecer um erro similar.

4.4.2 Conjunto de dados reais

O resultado para as simulação com conjuntos de dados reais são apresentados nessa seção. Para isso, dado um conjunto de dados, este foi dividido em outros dois, seguindo uma proporção de dois terços para treinamento e um terço para testes. A seleção de qual padrão pertence a qual conjunto foi feita de forma aleatória em cada um das 30 realizações.

As simulações foram realizadas com nove conjuntos de dados, a saber: Íris (IRI), Haberman (HAB), Patologias da Coluna Verterbral (PCV), Patologias da Derme (DER), Breast Cancer Wisconsin (BCW), Diabetes (DIA), Ripley (RIP), Spiral (SPI) e Banana (BAN) que constintuem o repositório *UCI Machine Learning*. Mais detalhes sobre a quantidade de padrões e dimensionalidade dos dados podem ser observadas na Tabela 10.

Nas tabelas 11, 12, 13 e 14 são apresentados os desempenhos, em termos de acurácia e do desvio padrão para as SVMC, LSSVC, P-LSSVC, IP-LSSVC e a proposta deste

trabalho. É apresentado também a quantidade média de vetores suporte (#SVs) e o tempo de treinamento e teste.

Uma vez que a OP-LSSVC e MP-LSSVC fazem uso de apenas uma pequena parcela de vetores suporte, foi escolhido inicialmente para o processo de otimização do *Leave-one-out* apenas os 100 melhores, selecionados pelo MRSR e MP. Essa metodologia só não foi seguida para o conjunto de dados (SPI) quando avaliadas com o *kernel* RBF, onde neste caso foram utilizados 900 vetores suporte inicialmente. Isso se fez necessário pois o desempenho da OP-LSSVC e da MP-LSSVC não encontravam-se adequados. Apesar disso, todos os outros classificadores que obteram quantidade semelhante de vetores suporte para esta base.

A fim de comparar a proposta de forma similar a utilizada pela sua versão voltada a problemas de regressão, o teste estatístico de Wilcoxon foi utilizado. Desta vez, utilizando a acurácia como amostra para a verificação do teste. Os resultados do teste são apresentados nas tabelas 11, 12, 13 e 14 considerando um nível de significância de 5%. Em cada célula da tabela é verificado se o resultado obtido pela OP-LSSVC é equivalente ao apresentado por essa célula. As células que apresentam um • apontam um resultado inferior por parte da OP-LSSVC. De maneira geral, pode-se concluir que o resultado obtido pela OP-LSSVC, em termos de acurácia, é equivalente ou superior a todas as outras técnicas, a saber: SVC, LSSVC, P-LSSVC, IP-LSSVC e MP-LSSVC. Essa análise só não se confirma em 9 resultados obtidos, 5 destes na abordagem linear da OP-LSSVC e outros 4 na abordagem com *kernel* RBF.

Tabela 11 – Resultado *kernel* Linear Parte 1 - Acurácia e desvio padrão após 30 execuções independentes utilizando *kernel* Linear. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.

Classificador		IRI	HAB	PCV	DER	BCW
SVM	ACC	100,0 ± 0,0	72,4 ± 3,3	• 87,1 ± 2,6	99,9 ± 0,3	96,4 ± 1,0
	#SVs	23,7 ± 0,9	97,4 ± 6,9	53,3 ± 6,3	22,9 ± 4,2	39,5 ± 10,9
	Tempo	4,5e-01±3,9e-01	9,6e+02±1,4e+03	2,5e+02±4,7e+02	9,6e-01±2,5e-01	2,3e+02±6,2e+02
LSSVM	ACC	100,0 ± 0,0	73,7 ± 3,2	83,5 ± 4,0	99,7 ± 0,5	96,1 ± 1,1
	#SVs	99,0 ± 0,0	201,0 ± 0,0	204,0 ± 0,0	241,0 ± 0,0	461,0 ± 0,0
	Tempo	1,6e+00±7,8e-01	6,6e+00±1,3e+00	5,6e+00±1,5e+00	8,0e+00±2,0e+00	4,0e+01±9,1e+00
Ip-LSSVM _{10%}	ACC	100,0 ± 0,0	70,6 ± 5,7	76,3 ± 6,4	99,2 ± 1,3	96,6 ± 0,8
	#SVs	89,0 ± 0,0	181,0 ± 0,0	184,0 ± 0,0	217,0 ± 0,0	415,0 ± 0,0
	Tempo	1,9e+00±9,9e-01	5,8e+00±2,4e+00	5,2e+00±2,2e+00	8,6e+00±3,5e+00	3,4e+01±8,4e+00
Ip-LSSVM _{20%}	ACC	100,0 ± 0,0	62,9 ± 9,1	67,9 ± 13,3	98,9 ± 1,7	• 96,8 ± 1,0
	#SVs	79,0 ± 0,0	161,0 ± 0,0	163,0 ± 0,0	193,0 ± 0,0	369,0 ± 0,0
	Tempo	1,7e+00±7,8e-01	5,3e+00±1,4e+00	4,7e+00±1,6e+00	7,8e+00±3,0e+00	3,6e+01±7,7e+00
Ip-LSSVM _{30%}	ACC	99,7 ± 0,9	60,2 ± 11,5	65,5 ± 15,6	98,6 ± 2,8	• 97,0 ± 1,0
	#SVs	69,0 ± 0,0	141,0 ± 0,0	143,0 ± 0,0	169,0 ± 0,0	323,0 ± 0,0
	Tempo	1,8e+00±8,4e-01	4,6e+00±1,7e+00	5,2e+00±1,9e+00	7,4e+00±2,8e+00	3,2e+01±8,4e+00
P-LSSVM	ACC	99,1 ± 3,6	72,2 ± 3,8	• 85,7 ± 3,0	98,5 ± 1,8	95,6 ± 2,5
	#SVs	2,7 ± 1,1	106,5 ± 19,1	131,7 ± 54,0	39,0 ± 58,1	38,7 ± 9,5
	Tempo	4,0e+01±1,1e+01	3,5e+01±8,5e+00	2,4e+01±2,0e+01	5,5e+01±1,8e+01	2,4e+02±2,9e+01
OP-LSSVM	ACC	100,0 ± 0,0	74,2 ± 3,1	83,9 ± 3,0	99,7 ± 0,5	95,8 ± 0,9
	#SVs	45,0 ± 0,0	83,0 ± 16,7	78,6 ± 12,8	89,8 ± 10,1	83,6 ± 13,0
	Tempo	8,1e+01±4,2e+01	1,4e+02±1,0e+02	1,3e+02±6,7e+01	1,6e+02±8,3e+01	5,8e+02±2,9e+02
MP-LSSVM	ACC	100,0 ± 0,0	74,2 ± 4,2	83,8 ± 3,6	99,7 ± 1,1	95,9 ± 1,1
	#SVs	60,0 ± 3,1	89,0 ± 17,8	98,6 ± 14,2	99,8 ± 12,3	93,5 ± 17,1
	Tempo	7,1e+01±4,8e+01	0,9e+02±1,9e+02	0,9e+02±7,3e+01	0,8e+02±7,2e+01	5,0e+02±3,5e+02

Tabela 12 – Resultado *kernel* Linear Parte 2 - Acurácia e desvio padrão após 30 execuções independentes utilizando *kernel* Linear. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.

Classificador		DIA	RIP	SPI	BAN
SVM	ACC	76,9 ± 2,3	88,0 ± 1,4	52,4 ± 2,2	55,1 ± 0,8
	#SVs	240,4 ± 14,4	226,6 ± 36,5	1135,4 ± 11,6	2834,9 ± 37,3
	Tempo	6,0e+03±8,6e+03	3,9e+02±9,2e+02	5,6e+02±1,2e+03	1,7e+04±2,2e+04
LSSVM	ACC	76,8 ± 2,1	87,9 ± 1,0	51,7 ± 2,0	54,8 ± 2,9
	#SVs	506,0 ± 0,0	825,0 ± 0,0	1320,0 ± 0,0	3498,0 ± 0,0
	Tempo	4,7e+01±7,8e+00	1,4e+02±1,8e+01	4,4e+02±6,0e+01	6,2e+03±9,7e+02
Ip-LSSVM _{10%}	ACC	73,5 ± 3,3	87,5 ± 2,3	50,1 ± 1,2	44,1 ± 3,7
	#SVs	455,0 ± 0,0	742,0 ± 0,0	1188,0 ± 0,0	3148,0 ± 0,0
	Tempo	4,3e+01±1,1e+01	1,3e+02±2,0e+01	4,1e+02±3,6e+01	5,6e+03±5,4e+02
Ip-LSSVM _{20%}	ACC	69,1 ± 10,0	87,2 ± 1,8	50,0 ± 1,8	44,9 ± 2,6
	#SVs	405,0 ± 0,0	660,0 ± 0,0	1056,0 ± 0,0	2798,0 ± 0,0
	Tempo	4,5e+01±1,3e+01	1,4e+02±2,2e+01	3,7e+02±3,1e+01	5,4e+03±7,1e+02
Ip-LSSVM _{30%}	ACC	53,1 ± 14,4	86,7 ± 1,6	50,5 ± 1,8	43,1 ± 1,6
	#SVs	354,0 ± 0,0	577,0 ± 0,0	924,0 ± 0,0	2449,0 ± 0,0
	Tempo	4,0e+01±1,1e+01	1,2e+02±1,4e+01	3,5e+02±3,0e+01	4,7e+03±6,1e+02
P-LSSVM	ACC	76,1 ± 2,4	87,3 ± 2,1	51,9 ± 2,1	• 57,0 ± 1,8
	#SVs	273,2 ± 59,8	224,9 ± 17,0	1150,0 ± 20,4	3029,5 ± 64,0
	Tempo	2,9e+02±8,0e+01	1,3e+03±1,1e+02	7,4e+02±2,8e+02	7,3e+03±2,3e+03
OP-LSSVM	ACC	76,4 ± 2,6	88,0 ± 1,3	51,7 ± 2,1	55,1 ± 2,7
	#SVs	82,8 ± 14,5	80,6 ± 17,8	90,4 ± 11,7	87,2 ± 15,3
	Tempo	6,5e+02±2,7e+02	1,6e+03±6,5e+02	3,8e+03±1,2e+03	2,8e+04±7,8e+03
MP-LSSVM	ACC	76,6 ± 2,8	88,1 ± 1,2	51,6 ± 2,2	54,8 ± 3,6
	#SVs	78,4 ± 14,5	85,7 ± 21,6	94,3 ± 14,3	92,2 ± 17,6
	Tempo	4,8e+02±1,8e+02	0,9e+03±7,3e+02	2,1e+03±1,2e+03	1,1e+04±6,7e+03

Tabela 13 – Resultado *kernel* RBF Parte 1 - Acurácia e desvio padrão após 30 execuções independentes utilizando *kernel* RBF. O resultados com um ponto preto na frete, (●), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.

Classificador		IRI	HAB	PCV	DER	BCW
SVM	ACC	100,0 ± 0,0	72,3 ± 3,6	86,4 ± 3,6	99,7 ± 0,5	96,4 ± 1,1
	#SVs	38,5 ± 1,5	110,0 ± 30,6	62,7 ± 14,6	45,8 ± 18,2	62,0 ± 31,1
	Tempo	1,9e+00±2,6e+00	3,1e+01±7,9e+01	1,5e+00±4,1e-01	1,7e+00±3,0e-01	2,1e+00±4,6e-01
LSSVM	ACC	98,8 ± 1,6	72,3 ± 3,7	86,1 ± 3,6	99,7 ± 0,6	96,4 ± 1,3
	#SVs	99,0 ± 0,0	201,0 ± 0,0	204,0 ± 0,0	241,0 ± 0,0	461,0 ± 0,0
	Tempo	2,7e+00±4,9e-01	7,4e+00±2,4e+00	8,5e+00±1,7e+00	1,5e+01±1,3e+00	4,4e+01±7,2e+00
Ip-LSSVM _{10%}	ACC	99,9 ± 0,0	72,9 ± 5,2	81,0 ± 11,9	98,9 ± 2,3	96,4 ± 1,4
	#SVs	89,0 ± 0,0	181,0 ± 0,0	184,0 ± 0,0	217,0 ± 0,0	415,0 ± 0,0
	Tempo	2,0e+00±9,9e-01	7,1e+00±2,0e+00	6,0e+00±2,5e+00	1,0e+01±4,6e-01	3,7e+01±6,9e+00
Ip-LSSVM _{20%}	ACC	99,9 ± 0,0	73,4 ± 4,0	82,8 ± 3,1	99,1 ± 2,0	95,7 ± 1,9
	#SVs	79,0 ± 0,0	161,0 ± 0,0	163,0 ± 0,0	193,0 ± 0,0	369,0 ± 0,0
	Tempo	1,8e+00±7,9e-01	7,2e+00±1,7e+00	7,3e+00±9,8e+00	1,0e+01±6,6e-01	3,7e+01±7,3e+00
Ip-LSSVM _{30%}	ACC	99,7 ± 0,9	72,1 ± 10,0	82,1 ± 11,9	98,0 ± 7,4	95,2 ± 3,1
	#SVs	69,0 ± 0,0	141,0 ± 0,0	143,0 ± 0,0	169,0 ± 0,0	323,0 ± 0,0
	Tempo	1,9e+00±8,6e-01	6,3e+00±1,4e+00	5,1e+00±3,1e-01	1,0e+01±3,5e+00	3,9e+01±1,6e+01
P-LSSVM	ACC	99,3 ± 0,9	71,6 ± 4,5	84,1 ± 4,4	98,6 ± 1,5	95,6 ± 1,9
	#SVs	6,3 ± 10,6	110,3 ± 22,8	77,8 ± 36,8	20,6 ± 34,3	54,5 ± 49,5
	Tempo	1,0e+02±2,3e+01	1,1e+02±3,5e+01	1,4e+02±5,8e+01	2,5e+02±5,3e+01	5,6e+02±1,3e+02
OP-LSSVM	ACC	99,9 ± 0,4	74,2 ± 3,7	86,7 ± 3,0	100,0 ± 0,0	96,2 ± 1,0
	#SVs	2,9 ± 0,7	10,0 ± 10,1	16,2 ± 6,4	80,4 ± 15,5	15,8 ± 5,7
	Tempo	7,9e+01±3,2e+01	1,4e+02±8,2e+01	1,2e+02±7,1e+01	1,4e+02±7,2e+01	5,1e+02±2,3e+02
MP-LSSVM	ACC	99,6 ± 0,5	74,2 ± 4,6	86,8 ± 3,2	99,2 ± 3,1	94,2 ± 4,1
	#SVs	99,0 ± 0,0	89,2 ± 9,2	95,6 ± 6,5	58,0 ± 14,1	90,8 ± 9,5
	Tempo	6,9e+01±2,2e+01	0,9e+02±6,2e+01	0,6e+02±7,4e+01	0,8e+02±6,1e+01	3,9e+02±1,9e+02

Tabela 14 – Resultado *kernel* RBF Parte 2 - Acurácia e desvio padrão após 30 execuções independentes utilizando *kernel* RBF. O resultados com um ponto preto na frete, (•), apontam um desempenho pior, em termos de acurácia, para a OP-LSSVC em relação à técnica indicada. Para todo os outros casos a OP-LSSVC é melhor ou equivalente.

Classificador		DIA	RIP	SPI	BAN
SVM	ACC	76,5 ± 2,4	• 90,5 ± 1,3	99,8 ± 0,2	90,2 ± 0,6
	#SVs	262,9 ± 24,7	179,5 ± 28,3	684,5 ± 73,6	786,4 ± 137,9
	Tempo	6,2e+00±1,3e+00	7,2e+01±2,1e+02	7,2e+02±1,8e+03	8,6e+02±2,7e+03
LSSVM	ACC	75,9 ± 2,4	• 90,5 ± 1,2	99,8 ± 0,2	90,2 ± 2,9
	#SVs	506,0 ± 0,0	825,0 ± 0,0	1320,0 ± 0,0	3498,0 ± 0,0
	Tempo	5,1e+01±7,3e+00	1,7e+02±1,9e+01	1,9e+03±7,4e+02	6,5e+03±9,6e+02
Ip-LSSVM _{10%}	ACC	75,6 ± 2,6	90,0 ± 1,1	97,2 ± 1,2	90,1 ± 3,6
	#SVs	455,0 ± 0,0	742,0 ± 0,0	1188,0 ± 0,0	3148,0 ± 0,0
	Tempo	4,9e+01±6,2e+00	1,6e+02±3,2e+01	4,2e+02±3,6e+01	5,7e+03±5,4e+02
Ip-LSSVM _{20%}	ACC	70,5 ± 8,3	90,4 ± 1,4	97,2 ± 1,8	90,0 ± 2,7
	#SVs	405,0 ± 0,0	660,0 ± 0,0	1056,0 ± 0,0	2798,0 ± 0,0
	Tempo	4,5e+01±4,9e+00	1,4e+02±2,9e+01	3,8e+02±3,1e+01	5,8e+03±7,2e+02
Ip-LSSVM _{30%}	ACC	62,4 ± 7,9	90,3 ± 0,9	88,9 ± 1,8	88,7 ± 1,4
	#SVs	354,0 ± 0,0	577,0 ± 0,0	924,0 ± 0,0	2449,0 ± 0,0
	Tempo	4,8e+01±1,4e+01	1,4e+02±3,2e+01	3,6e+02±3,0e+01	5,1e+03±6,2e+02
P-LSSVM	ACC	• 76,6 ± 2,3	89,6 ± 1,5	98,2 ± 1,1	89,9 ± 0,9
	#SVs	279,7 ± 46,6	236,6 ± 76,0	334,0 ± 222,3	813,0 ± 342,1
	Tempo	5,3e+02±1,5e+02	2,1e+03±3,0e+02	6,9e+03±1,7e+03	9,1e+03±2,6e+03
OP-LSSVM	ACC	75,4 ± 2,2	89,5 ± 1,3	99,6 ± 0,3	90,2 ± 0,6
	#SVs	22,3 ± 3,4	20,2 ± 5,0	801,0 ± 87,1	97,0 ± 9,6
	Tempo	5,6e+02±2,2e+02	1,5e+03±5,8e+02	3,4e+03±9,7e+02	2,6e+04±6,7e+03
MP-LSSVM	ACC	75,5 ± 3,6	• 90,5 ± 1,1	98,3 ± 1,9	88,3 ± 0,9
	#SVs	97,0 ± 3,8	86,0 ± 12,4	801,0 ± 78,0	97,0 ± 15,0
	Tempo	4,6e+02±1,9e+02	0,9e+03±6,7e+02	2,6e+03±8,4e+02	2,4e+04±6,8e+03

Nas tabelas 15 e 16 são apresentados o percentual de redução em relação a quantidade de vetores suporte utilizados pela LSSVC. A OP-LSSVC obteve uma redução entre 54, 5% (IRI) e 97, 5% (BAN) para *kernel* lineares. Em relação a *kernel* RBF, a OP-LSSVC obteve um redução entre 39, 3% (SPI) e 97, 2% (BAN). Em termos de esparsidade, a OP-LSSVC apresentou o menor modelo na maioria dos conjuntos de dados, principalmente com *kernel* RBF onde apenas dois resultados não foram os melhores entre as outras técnicas. Nos resultados com *kernel* linear os resultados foram os melhores para as seguintes bases: HAB (58,7%), RIP (90,2%), SPI (93,2%) e BAN (97,5%).

Tabela 15 – Percentual de redução entre o classificador apresentado em cada linha e a LSSVM com *kernel* linear.

Classifcador	IRI	HAB	PCV	DER	BCW	DIA	RIP	SPI	BAN
SVM	78.1	51.5	73.9	90.5	91.4	52.5	72.5	14.0	19.0
LSSVM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IP-LSSVM 10%	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
IP-LSSVM 20%	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
IP-LSSVM 30%	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
P-LSSVM	97.2	47.2	35.7	83.8	91.6	46.0	72.8	12.8	13.4
OP-LSSVM	54.5	58.7	61.5	62.7	81.9	83.6	90.2	93.2	97.5
MP-LSSVM	39.4	55.7	51.7	58.6	79.7	84.5	89.6	92.9	97.4

Tabela 16 – Percentual de redução entre o regressor apresentado em cada linha e a LSSVM com *kernel* RBF.

Classifcador	IRI	HAB	PCV	DER	BCW	DIA	RIP	SPI	BAN
SVM	61.1	45.3	69.3	81.0	86.6	48.0	78.2	48.1	77.5
LSSVM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IP-LSSVM 10%	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
IP-LSSVM 20%	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
IP-LSSVM 30%	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
P-LSSVM	93.6	45.3	61.9	91.5	88.2	44.9	71.4	74.7	76.8
OP-LSSVM	97.1	95.0	92.1	66.6	96.6	95.6	97.6	39.3	97.2
MP-LSSVM	0.0	55.6	53.1	75.9	80.3	80.8	89.5	39.3	97.2

5 Conclusão e Trabalhos Futuros

As máquinas de vetores suportes de mínimos quadrados, apresentada anteriormente neste trabalho, tornaram-se o estado da arte no que se refere às técnicas de classificação e regressão. Na formulação da LSSVM, em vez de resolver um problema de programação quadrática com restrições lineares de desigualdade, a LSSVM utiliza a norma L2 como função de erro. Isso simplifica o problema dual de otimização à solução de um problema linear. Um problema adquirido ao realizar essa simplificação é a falta de esparsidade do modelo obtido pela LSSVM.

Foi apresentada uma alternativa para resolver o problema de falta de esparsidade em máquinas de vetores suportes quadradas, a OP-LSSVM e a MP-LSSVM. A ideia deste trabalho é reduzir o número de padrões candidatos a serem vetores suporte. Isso foi alcançado realizando a redução da dimensionalidade do espaço de características com a utilização dos métodos de regressão esparsa *Multiresponse Sparse Regression* ou *Matching Pursuit* combinados com a validação cruzada *leave-one-out*.

A proposta consiste em selecionar os vetores mais relevantes utilizando o MRSR e, logo após isso, o melhor subconjunto de vetores é escolhido com o auxílio de uma técnica de validação cruzada. Baseado nos resultados obtidos pode-se concluir que o objetivo de obter um método de regressão e classificação esparsos foi atingido. Além disso, a presente proposta é uma alternativa interessante em relação a outras técnicas uma vez que os resultados obtidos são melhores ou equivalentes. Em relação ao tempo de treinamento, pode-se verificar que à medida que o conjunto de dados aumenta de tamanho, os tempos de treinamento da LSSVM e da OP-LSSVM se aproximam de maneira a indicar que, possivelmente, a OP-LSSVM vai ter o tempo de treinamento menor. Apesar de um desempenho inferior, em relação a OP-LSSVM, a MP-LSSVM apresentou resultados satisfatórios em problemas de regressão e principalmente de classificação, sendo muitas vezes equivalentes à OP-LSSVM.

Como trabalho futuro, pode-se explorar a possibilidade de obtenção de conjuntos reduzidos utilizando outras técnicas de seleção de regressores lineares iterativos de segunda ordem como Levenberg-Maquardt e gradiente descendente. Técnicas iterativas permitem que seja aplicado, como alternativa, um critério de parada durante o treinamento, como por exemplo, o erro quadrático médio obtido ao inserir cada novo vetor.

Uma vez que a OP-LSSVM faz um extenso uso de operações matriciais básicas como inversões e produtos de matrizes, estas podem ser realizadas de maneira mais rápida se aplicadas em hardwares preparados para esta atividade, como unidades gráficas de processamento (Graphics Processing Unit, GPU).

Referências

- ALLEN, D. M. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, [Taylor Francis, Ltd., American Statistical Association, American Society for Quality], v. 16, n. 1, p. 125–127, 1974. ISSN 00401706. Disponível em: <<http://www.jstor.org/stable/1267500>>. Citado na página 56.
- ANCHALIA, P. P.; ROY, K. Two class fisher's linear discriminant analysis using mapreduce. In: *2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim)*. [S.l.: s.n.], 2015. p. 463–469. Citado na página 18.
- AURIA, R. A. M. L. Support vector machines (svm) as a technique for solvency analysis. *DIW Berlin Discussion Paper*, aug 2008. Citado na página 18.
- BEN-ISRAEL, T. N. G. A. *Generalized Inverses: Theory and Applications*. 2nd. ed. Springer, 2003. ISBN 0387002936,9780387216348,9780387002934. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=01B4708D8B17672C010846BEA6C5DE9A>>. Citado na página 40.
- Bertsimas, D.; King, A.; Mazumder, R. Best Subset Selection via a Modern Optimization Lens. *ArXiv e-prints*, jul. 2015. Citado na página 46.
- BLAIR, J. J. H. R. C. A comparison of the power of wilcoxon's rank-sum statistic to that of student's t statistic under various nonnormal distributions. *Journal of Educational Statistics*, [Sage Publications, Inc., American Educational Research Association, American Statistical Association], v. 5, n. 4, p. 309–335, 1980. ISSN 03629791. Disponível em: <<http://www.jstor.org/stable/1164905>>. Citado na página 61.
- BRABANTER, K. D. *et al.* Optimized fixed-size kernel models for large data sets. *Comput. Stat. Data Anal.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 54, n. 6, p. 1484–1504, jun. 2010. ISSN 0167-9473. Disponível em: <<http://dx.doi.org/10.1016/j.csda.2010.01.024>>. Citado na página 20.
- BURGES, C. J. Simplified support vector decision rules. In: . [S.l.]: Morgan Kaufmann, 1996. p. 71–77. Citado 2 vezes nas páginas 19 e 20.
- CAI, T. T.; WANG, L. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, v. 57, n. 7, p. 4680–4688, July 2011. ISSN 0018-9448. Citado na página 48.
- CARVALHO, B. P. R.; BRAGA, A. P. IP-LSSVM: A two-step sparse classifier. *Pattern Recognition Letters*, v. 30, p. 1507–1515, 2009. Citado 2 vezes nas páginas 20 e 41.
- CUNNINGHAM, J. P.; GHAMRANI, Z. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, v. 16, p. 2859–2900, 2015. Disponível em: <<http://jmlr.org/papers/v16/cunningham15a.html>>. Citado na página 45.
- D'AMATO, L.; MORENO, J. A.; MUJICA, R. Reducing the complexity of kernel machines with neural growing gas in feature space. In: *Proceedings of the IX (IBERAMIA'04)*. [S.l.]: Springer, 2004. LNAI-3315, p. 799–808. Citado na página 21.

- DOWNS, T.; GATES, K. E.; MASTERS, A. Exact simplification of support vector solutions. *Journal of Machine Learning Research*, v. 2, p. 293–297, 2002. Citado na página 20.
- EFRON, B. *et al.* Least angle regression. *Annals of Statistics*, v. 32, p. 407–499, 2004. Citado na página 46.
- EFRON, B. *et al.* Least angle regression. *Ann. Statist.*, The Institute of Mathematical Statistics, v. 32, n. 2, p. 407–499, 04 2004. Disponível em: <<http://dx.doi.org/10.1214/009053604000000067>>. Citado na página 52.
- FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, v. 7, n. 7, p. 179–188, 1936. Citado na página 18.
- FOUSSIER, P. *From Product Description to Cost: A Practical Approach: Volume 1: The Parametric Approach*. Springer, 2006. (Decision Engineering). ISBN 9781846283352. Disponível em: <<https://books.google.com.br/books?id=P3pRMDleQo8C>>. Citado na página 57.
- FUKUNAGA, K. *Introduction to Statistical Pattern Recognition (2Nd Ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990. ISBN 0-12-269851-7. Citado na página 45.
- FURNIVAL, R. W. W. G. M. Regressions by leaps and bounds. *Technometrics*, [Taylor Francis, Ltd., American Statistical Association, American Society for Quality], v. 16, n. 4, p. 499–511, 1974. ISSN 00401706. Disponível em: <<http://www.jstor.org/stable/1267601>>. Citado na página 46.
- GEEBELLEN, 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 Networks and Learning Systems*, v. 23, n. 4, p. 682–688, April 2012. ISSN 2162-237X. Citado na página 20.
- GRIGELIONIS, B. *Student's t-Distribution and Related Stochastic Processes*. 1. ed. Springer-Verlag Berlin Heidelberg, 2013. (SpringerBriefs in Statistics). ISBN 978-3-642-31145-1, 978-3-642-31146-8. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=4B950087FDD199F57A8381CC049C0530>>. Citado na página 61.
- GUYON STEVE GUNN, M. N. L. A. Z. I. *Feature Extraction: Foundations and Applications*. Springer, 2006. (Studies in Fuzziness and Soft Computing). ISBN 3540354875, 9783540354871. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=A623C492DF3719EFEDD45FB53E726767>>. Citado na página 50.
- HAMEED, M. A. *COMPARATIVE ANALYSIS OF ORTHOGONAL MATCHING PURSUIT AND LEAST ANGLE REGRESSION*. Dissertação (Mestrado) — Michigan State University, 1 2012. Citado na página 44.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics). Citado na página 48.
- HOAGLIN, D. C.; WELSCH, R. E. The hat matrix in regression and ANOVA. *The American Statistician*, v. 32, n. 1, p. 17–22, 1978. Citado na página 56.

HUSSAIN, A. *et al.* Reduced set support vector machines: Application for 2-dimensional datasets. In: *Proceedings of the 2nd International Conference on Signal Processing and Communication Systems (ICSPCS'08)*. [S.l.: s.n.], 2008. p. 1–4. Citado na página 20.

KARSMARKERS, P. *et al.* Sparse conjugate directions pursuit with application to fixed-size kernel models. *Machine Learning*, v. 85, n. 1, p. 109–148, 2011. ISSN 1573-0565. Disponível em: <<http://dx.doi.org/10.1007/s10994-011-5253-8>>. Citado na página 21.

KEERTHI, S. S.; CHAPELLE, O.; DECOSTE, D. Building support vector machines with reduced classifier complexity. *J. Mach. Learn. Res.*, JMLR.org, v. 7, p. 1493–1515, dez. 2006. ISSN 1532-4435. Disponível em: <<http://dl.acm.org/citation.cfm?id=1248547.1248602>>. Citado na página 21.

KOGGALAGE, R.; HALGAMUGE, S. Reducing the number of training samples for fast support vector machine classification. In: . [S.l.: s.n.], 2004. Citado na página 19.

LAY, D. C. *Linear Algebra and Its Applications (3rd Edition)*. [S.l.]: Addison Wesley, 2002. Citado na página 20.

LEE, D. Q. *NUMERICALLY EFFICIENT METHODS FOR SOLVING LEAST SQUARES PROBLEMS*. 2012. Citado na página 19.

LIANG, X. An effective method of pruning support vector machine classifiers. *Trans. Neur. Netw.*, IEEE Press, Piscataway, NJ, USA, v. 21, n. 1, p. 26–38, jan. 2010. ISSN 1045-9227. Disponível em: <<http://dx.doi.org/10.1109/TNN.2009.2033677>>. Citado na página 20.

LU, S. X.; MENG, J.; CAO, G. E. Support vector machine based on a new reduced samples method. In: *2010 International Conference on Machine Learning and Cybernetics*. [S.l.: s.n.], 2010. v. 3, p. 1510–1514. ISSN 2160-133X. Citado na página 19.

MALLAT, S. *A wavelet tour of signal processing the Sparse way*. 3. ed. Academic Press, 2008. v. 1. (1, v. 1). ISBN 9780123743701,0123743702. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=2DC545A53DB15C8557022C7F978A57CC>>. Citado 2 vezes nas páginas 46 e 47.

MALLAT, S. G.; ZHANG, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, v. 41, n. 12, p. 3397–3415, Dec 1993. ISSN 1053-587X. Citado 2 vezes nas páginas 46 e 47.

MEHRKANOON, S. *et al.* Multiclass semisupervised learning based upon kernel spectral clustering. *IEEE Trans. Neural Netw. Learning Syst.*, v. 26, n. 4, p. 720–733, 2015. Disponível em: <<http://dx.doi.org/10.1109/TNNLS.2014.2322377>>. Citado na página 20.

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. Series A, Containing Papers of a Mathematical or Physical Character*, The Royal Society, v. 209, p. 415–446, 1909. ISSN 02643952. Disponível em: <<http://www.jstor.org/stable/91043>>. Citado na página 35.

MICHE, Y. *et al.* Op-elm: Optimally pruned extreme learning machine. *Neural Networks, IEEE Transactions on*, v. 21, n. 1, p. 158–162, Jan 2010. Citado na página 57.

- MYERS, R. *Classical and Modern Regression with Applications*. [S.l.]: PWS-KENT, 1990. (Bookware Companion Series). ISBN 9780534921781. Citado 2 vezes nas páginas 56 e 57.
- NAGY, G. State of the art in pattern recognition. *Proceedings of the IEEE*, v. 56, n. 5, p. 836–863, May 1968. ISSN 0018-9219. Citado na página 18.
- NATARAJAN, B. K. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, v. 24, n. 2, p. 227–234, 1995. Disponível em: <<http://dx.doi.org/10.1137/S0097539792240406>>. Citado na página 46.
- PLATT, J. C. Fast training of support vector machines using sequential minimal optimization. In: *Advances in Kernel Methods - Support Vector Learning*. Cambridge, MA, USA: MIT Press, 1999. Citado 2 vezes nas páginas 19 e 30.
- PRENTER, P. M. The numerical treatment of integral equations (c. t. h. baker). *SIAM Review*, Society for Industrial & Applied Mathematics (SIAM), v. 23, n. 2, p. 266–267, apr 1981. Disponível em: <<https://doi.org/10.1137%2F1023054>>. Citado na página 20.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 65–386, 1958. Citado na página 18.
- SCHOLKOPF, B. *et al.* Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, v. 10, n. 5, p. 1000–1017, Sep 1999. ISSN 1045-9227. Citado na página 20.
- SCHÖLKOPF, B. *et al.* New support vector algorithms. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 12, n. 5, p. 1207–1245, maio 2000. ISSN 0899-7667. Disponível em: <<http://dx.doi.org/10.1162/089976600300015565>>. Citado na página 31.
- SHERMAN, J.; MORRISON, W. J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Statist.*, The Institute of Mathematical Statistics, v. 21, n. 1, p. 124–127, 03 1950. Disponível em: <<http://dx.doi.org/10.1214/aoms/1177729893>>. Citado na página 89.
- SILVA, J. P.; NETO, A. R. d. R. Sparse least squares support vector machines via genetic algorithms. In: *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*. [S.l.: s.n.], 2013. p. 248–253. ISSN 2377-0589. Citado na página 21.
- SIMILÄ, T.; TIKKA, J. Multiresponse sparse regression with application to multidimensional scaling. In: *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*. [S.l.]: Springer Berlin Heidelberg, 2005, (LNCS, v. 3697). p. 97–102. Citado na página 52.
- STURM, B. L.; CHRISTENSEN, M. G. Cyclic matching pursuits with multiscale time-frequency dictionaries. In: *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*. [S.l.: s.n.], 2010. p. 581–585. ISSN 1058-6393. Citado na página 47.
- 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*. [S.l.: s.n.], 2000. v. 2, p. 757–760 vol.2. Citado 2 vezes nas páginas 20 e 39.

SUYKENS, J. A. K.; LUKAS, L.; VANDEWALLE, J. Sparse least squares support vector machine classifiers. In: *Proceedings of the 8th European Symposium on Artificial Neural Networks (ESANN'00)*. [S.l.: s.n.], 2000. p. 37–42. Citado 2 vezes nas páginas 20 e 40.

SUYKENS, J. A. K.; VANDEWALLE, J. Least squares support vector machine classifiers. *Neural Processing Letters*, v. 9, n. 3, p. 293–300, 1999. Citado 2 vezes nas páginas 19 e 36.

SUYKENS T. VAN GESTEL, J. D. B. B. D. M. J. V. J. A. K. *Least Squares Support Vector Machines* (ISBN 981-238-151-1). [S.l.]: Publisher, 2002. Citado 2 vezes nas páginas 20 e 21.

TANG, B.; MAZZONI, D. Multiclass reduced-set support vector machines. In: *Proceedings of the 23rd International Conference on Machine Learning (ICML'2006)*. [S.l.: s.n.], 2006. p. 921–928. Citado na página 20.

TARPEY, T. A note on the prediction sum of squares statistic for restricted least squares. *The American Statistician*, [American Statistical Association, Taylor Francis, Ltd.], v. 54, n. 2, p. 116–118, 2000. ISSN 00031305. Disponível em: <<http://www.jstor.org/stable/2686028>>. Citado na página 56.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York: Springer, 1995. Citado na página 18.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. 2nd. ed. Springer, 2000. (Statistics for Engineering and Information Science). ISBN 1441931600, 9781441931603. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=BC6C3185972183559646187BE34C1C31>>. Citado na página 30.

WEISBERG, S. *Applied Linear Regression*. Wiley, 2005. (Wiley Series in Probability and Statistics). ISBN 9780471704089. Disponível em: <<https://books.google.com.br/books?id=xd0tNdFOOjcC>>. Citado na página 46.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, [International Biometric Society, Wiley], v. 1, n. 6, p. 80–83, 1945. ISSN 00994987. Disponível em: <<http://www.jstor.org/stable/3001968>>. Citado 2 vezes nas páginas 61 e 66.

WOODBURY, M. A. Inverting Modified Matrices. In: _____. Princeton, NJ: Princeton University, 1950. (Statistical Research Group Memorandum Reports, 42). Citado na página 89.

YANG, J.; BOUZERDOUM, A.; PHUNG, S. L. A training algorithm for sparse LS-SVM using compressive sampling. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19 March 2010, Sheraton Dallas Hotel, Dallas, Texas, USA*. [s.n.], 2010. p. 2054–2057. Disponível em: <<http://dx.doi.org/10.1109/ICASSP.2010.5495015>>. Citado na página 54.

YANG, W. F. Study on classification performance and parameter optimization of least squares support vector machine. *IJACT*, International Journal of Advancements in Computing Technology, v. 4, n. 12, 07 2012. Disponível em: <<http://dx.doi.org/10.1214/aoms/1177729893>>. Citado na página 60.

YANG YULING WANG, Y. W. F. Study on classification performance and parameter optimization of least squares support vector machine. *International Journal of Advancements in Computing Technology(IJACT)*, v. 4, n. 12, p. 152–129, jul 2012. Citado na página 60.

ZHANG, W.; NIU, P. Ls-svm based on chaotic particle swarm optimization with simulated annealing and application. In: *2011 2nd International Conference on Intelligent Control and Information Processing*. [S.l.: s.n.], 2011. v. 2, p. 931–935. Citado na página 21.

Apêndices

APÊNDICE A – Equivalência entre o PRESS e o Leave-one-out

A equivalência entre o PRESS e o cômputo do *Leave-one-out* pode ser verificada através do auxílio de duas identidades matriciais. A identidade de Woodbury (1950), Sherman e Morrison (1950) e o produto de duas matrizes com adição ou remoção de linhas.

Para a primeira identidade considere as matrizes $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ e $\mathbf{D}$. A identidade matricial de Woodbury, por vezes chamada também de fórmula de Sherman-Morrison-Woodbury provê um resultado para a inversão da soma de matrizes na forma: $(\mathbf{A} + \mathbf{BCD})$. Sua identidade é apresentada na Equação A.1.

$$\begin{aligned} (\mathbf{A} + \mathbf{BCD})^{-1} &= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \\ (\mathbf{A} + \mathbf{BCD})^{-1} &= \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{BDA}^{-1}}{\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B}} \end{aligned} \quad (\text{A.1})$$

A fim de simplificar a demonstração considere que $\mathbf{B} = \mathbf{b}_i$ um vetor $(m \times 1)$, $\mathbf{D} = \mathbf{b}_i^T$ e $\mathbf{C} = 1$. Assim tem-se:

$$(\mathbf{A} + \mathbf{b}_i\mathbf{b}_i^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{b}_i\mathbf{b}_i^T\mathbf{A}^{-1}}{1 + \mathbf{b}_i^T\mathbf{A}^{-1}\mathbf{b}_i} \quad (\text{A.2})$$

Para a segunda identidade o produto $\mathbf{D}_{+j}^T\mathbf{D}_{+j}$, resultante da adição de uma linha à matriz $\mathbf{D}$ pode ser computado da seguinte maneira:

$$\mathbf{D}_{+j}^T\mathbf{D}_{+j} = \sum \mathbf{D}^T \quad \sum \mathbf{r}_j^T \times \begin{bmatrix} \mathbf{D} \\ \mathbf{r}_j \end{bmatrix} = \sum \mathbf{D}^T\mathbf{D} + \mathbf{r}_j^T\mathbf{r}_j \quad (\text{A.3})$$

portanto a remoção de uma das linhas da matriz $\mathbf{D}$ pode ser computada como:

$$\mathbf{D}_{-j}^T\mathbf{D}_{-j} = \mathbf{D}^T\mathbf{D} - \mathbf{r}_j^T\mathbf{r}_j \quad (\text{A.4})$$

De posse de ambas as identidades é possível obter o erro *leave-one-out*, para isso, considere o modelo $\mathbf{Dz} = \mathbf{t}$. Estima-se a matriz $\mathbf{z}_{-j}$ obtida a partir da base de dados $\mathbf{D}$ sem a amostra j , $\mathbf{D}_{-j}$. Assim tem-se:

$$\mathbf{D}_{-j}\mathbf{z}_{-j} = \mathbf{t}_{-j} \quad (\text{A.5})$$

$$\mathbf{z}_{-j} = (\mathbf{D}_{-j}^T\mathbf{D}_{-j})^{-1}\mathbf{D}_{-j}^T\mathbf{t}_{-j} \quad (\text{A.6})$$

Estimando $(\mathbf{D}_{-j}^\top \mathbf{D}_{-j})^{-1}$.

$$(\mathbf{D}_{-j}^\top \mathbf{D}_{-j})^{-1} = (\mathbf{D}^\top \mathbf{D} - \mathbf{r}_j^\top \mathbf{r}_j)^{-1} \quad (\text{A.7})$$

Substituindo $(\mathbf{D}^\top \mathbf{D} - \mathbf{r}_j^\top \mathbf{r}_j)$ na equação A.2

$$(\mathbf{D}^\top \mathbf{D} - \mathbf{r}_j^\top \mathbf{r}_j)^{-1} = (\mathbf{D}^\top \mathbf{D})^{-1} + \frac{(\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{r}_j^\top \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1}}{1 - \mathbf{r}_j^\top (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{r}_j} \quad (\text{A.8})$$

$$(\mathbf{D}_{-j}^\top \mathbf{D}_{-j})^{-1} = \frac{(1 - h_{jj})(\mathbf{D}^\top \mathbf{D})^{-1} + (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{r}_j^\top \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1}}{(1 - h_{jj})} \quad (\text{A.9})$$

Assim tem-se de acordo com a Equação A.6:

$$\mathbf{z}_{-j} = \frac{\sum (1 - h_{jj})(\mathbf{D}^\top \mathbf{D})^{-1} + (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{r}_j^\top \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \sum}{(1 - h_{jj})} \mathbf{D}_{-j}^\top \mathbf{t}_{-j} \quad (\text{A.10})$$

O erro leave-one-out pode ser estimado como:

$$e_{loo}^j = t_j - \hat{t}_{-j} \quad (\text{A.11})$$

$$e_{loo}^j = t_j - \mathbf{r}_j \frac{\sum (1 - h_{jj})(\mathbf{D}^\top \mathbf{D})^{-1} + (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{r}_j^\top \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \sum}{(1 - h_{jj})} \mathbf{D}_{-j}^\top \mathbf{t}_{-j} \quad (\text{A.12})$$

$$e_{loo}^j = t_j - \frac{\mathbf{r}_j (1 - h_{jj})(\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j} + \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{r}_j^\top \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j}}{(1 - h_{jj})} \quad (\text{A.13})$$

Substituindo o valor de h_{jj} e introduzindo o valor de t_j no numerador da equação, tem-se:

$$e_{loo}^j = \frac{(1 - h_{jj}) t_j - \mathbf{r}_j (1 - h_{jj})(\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j} + h_{jj} \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j}}{(1 - h_{jj})} \quad (\text{A.14})$$

Colocando o termo $(\mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j})$ em evidência e simplificando a equação, tem-se:

$$e_{loo}^j = \frac{(1 - h_{jj}) t_j - ((1 - h_{jj}) + h_{jj})(\mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j})}{(1 - h_{jj})} \quad (\text{A.15})$$

$$e_{loo}^j = \frac{(t_j - h_{jj} t_j) - (1 - h_{jj})(\mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j})}{(1 - h_{jj})} \quad (\text{A.16})$$

$$e_{loo}^j = \frac{(t_j - h_{jj} t_j) - (1 - h_{jj})(\mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j})}{(1 - h_{jj})} \quad (\text{A.17})$$

$$e_{loo}^j = \frac{(t_j - h_{jj} t_j) - (1 - h_{jj})(\mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}_{-j}^\top \mathbf{t}_{-j})}{(1 - h_{jj})} \quad (\text{A.18})$$

Substituindo $\mathbf{D}_{-j}^\top \mathbf{t}_{-j}$ por $\mathbf{D}^\top \mathbf{t} - \mathbf{r}_j^\top t_j$

$$e_{loo}^j = \frac{(t_j - h_{jj} t_j) - \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} (\mathbf{D}^\top \mathbf{t} - \mathbf{r}_j^\top t_j)}{(1 - h_{jj})} \quad (\text{A.19})$$

$$e_{loo}^j = \frac{(t_j - h_{jj} t_j) - \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}^\top \mathbf{t} + \mathbf{r}_j (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{r}_j^\top t_j}{(1 - h_{jj})} \quad (\text{A.20})$$

$$e_{loo}^j \stackrel{\text{XS}}{=} \frac{\hat{t}_j (t_j - h_{jj} t_j) - \mathbf{r}_j \mathbf{Z} + h_{jj} t_j}{(1 - h_{jj})} \quad (\text{A.21})$$

$$e_{loo}^j = \frac{t_j - \hat{t}_j}{(1 - h_{jj})} \quad (\text{A.22})$$

Na Equação A.22 é apresentada a forma equivalente para o cálculo do erro *Leave-one-out* quando uma amostra j é removida. O cômputo final do PRESS é feito através da Equação A.23.

$$PRESS = \sum_{j=1}^N e_{loo}^j. \quad (\text{A.23})$$