

## Um novo método de poda iterativo de máquinas de vetores-suporte de mínimos quadrados

Davi A. Leão\* Ajalmar R. Rocha Neto\*\*

\* Instituto Federal de Educação, Ciência e Tecnologia, CE, (e-mail: davi.alves.leao61@aluno.ifce.edu.br).

\*\* Instituto Federal de Educação, Ciência e Tecnologia, CE (e-mail: ajalmar@ifce.edu.br)

---

**Abstract:** Least-Square Support Vector Machines (LSSVM) are solved from a linear system, unlike support vector machines (SVM), which require the application of quadratic programming. So it is preferable in some problems to use LSSVM because the complexity of the resolution will be lower. However, unlike SVM, LSSVM generates non-sparse solutions. A significant disadvantage, given that all training patterns will be support vectors of LSSVM. To overcome such non-sparse solutions, we propose an iterative solution with pruning without causing a loss of efficiency, named as FSLM-LSSVM. The proposed method is compared with two others in the literature, P-LSSVM and IP-LSSVM. Finally, the evaluation of the proposed solution consisted of comparing the averages of the hit rates.

**Resumo:** As máquinas vetores-suporte de mínimos quadrados (Least Square Support Vector Machines) são resolvidas a partir de um sistema linear, ao contrário das máquinas vetores-suporte (Support Vector Machines), que é necessário a aplicação de programação quadrática. Com isso, é preferível, em alguns problemas, a utilização do LSSVM, uma vez que a complexidade de resolução será menor. Entretanto, ao contrário do SVM, o LSSVM gera soluções não esparsas, e essa é uma desvantagem significativa, dado que todos os padrões de treinamento serão vetores-suporte do LSSVM. Para superar tais soluções não esparsas, propomos uma solução iterativa com poda, baseada no método de Levenberg-Marquardt, sem causar perda de eficiência, doravante denominado FSLM-LSSVM. O método proposto, é comparado com outros dois da literatura, P-LSSVM e IP-LSSVM. Por fim, a avaliação da solução proposta consistiu na comparação das médias das taxas de acerto.

*Keywords:* LSSVM, Levenberg–Marquardt, Pruning, Sparseness, Optimization

*Palavras-chaves:* LSSVM, Levenberg–Marquardt, Poda, Esparsidade, Otimização

---

### 1. INTRODUÇÃO

As máquinas de vetores-suporte (*Support Vector Machines* - SVM), devido à sua forte base matemática, têm sido aplicadas a diversos tipos de problema, tanto de classificação binária quanto de regressão. O método SVM apresenta propriedades vantajosas quando comparado a outros modelos de aprendizado de máquina. Uma delas é a geração do hiperplano de decisão com base em um número pequeno de exemplos de entrada, denominados vetores-suporte (*Support Vector* - SV). Em outras palavras, são geradas soluções esparsas. O problema dual do SVM é normalmente solucionado por programação quadrática (*Quadratic programming* - QP) (Nocedal and Wright, 2006).

O método das máquinas de vetor-suporte de mínimos quadrados (*Least Square Support Vector Machine* - LSSVM) é um algoritmo de aprendizado supervisionado derivado do SVM tradicional. O LSSVM é capaz de resolver tanto problemas de classificação quanto de regressão. O Treinamento desse modelo é feito através da resolução de um sistema linear Karush-Kuhn-Tucker (KKT) (Fletcher, 1987), uma vantagem em relação ao SVM, dado que a solução de um sistema linear possui custo assintótico menor quando

comparado ao de programação quadrática. Em contrapartida, o LSSVM apresenta soluções não esparsas, ou seja, ao final do treinamento todos os exemplos de entrada são caracterizados como vetores-suporte. Desse modo, é necessário o armazenar todo o conjunto de treinamento e todos os multiplicadores de Lagrange associados para computar a função discriminante, situação essa indesejável em problemas com grandes quantidades de dados.

A fim de construir soluções esparsas com o LSSVM, foram propostos vários métodos para diminuição de complexidade baseados na redução dos vetores-suporte, podendo citar os métodos de poda. Os métodos de poda compreendem um conjunto de técnicas que visam simplificar a estrutura interna dos classificadores LSSVM, sem que o mesmo perca a sua eficiência. A ideia principal sobre os métodos de poda é detalhada por Burges (1996).

Para aplicar os métodos de poda é necessário obter os multiplicadores de Lagrange. Logo, os algoritmos que aplicam o método da pseudo inversa para resolver o sistema linear, são treinados pelo menos duas vezes. Com o objetivo de solucioná-lo e, simultaneamente, resolver o problema da esparsidade sem a necessidade de retrainar,

foi proposto neste trabalho a aplicação do método de Levenberg-Marquardt, um algoritmo de otimização de segunda-ordem, com a adição de uma poda iterativa. Essa proposta será descrita em detalhes na seção 5.

O presente trabalho encontra-se dividido em sete seções, na qual a primeira é a Introdução. Na seção 2 são descritos em detalhes os conceitos do LSSVM. Em seguida, na seção 3, são mostrados dois algoritmos da literatura que serão comparados com o proposto. Após isso, na seção 4, é apresentado o método de Levenberg-Marquardt. Na seção 5 está a descrição da proposta, com os resultados das simulações computacionais na seção 6. Por fim, as conclusões são explanadas na seção 7.

## 2. CLASSIFICADOR LSSVM

Dado um conjunto  $\{\mathbf{x}_i, y_i\}_{i=1}^N$ , no qual  $\mathbf{x}_i$  é o  $i$ -ésimo vetor de características e  $y_i$  é o rótulo associado. A formulação do problema primal para o LSSVM de Suykens and Vandewalle (1999) é dada por

$$\min_{\mathbf{w}, \xi_i, b} \left\{ \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^N \xi_i^2 \right\} \quad (1)$$

s.a  $y_i[\mathbf{w}^\top \mathbf{x}_i + b] = 1 - \xi_i, i = 1, \dots, N$

no qual  $\gamma$  é um termo de regularização da margem,  $\{\xi_i\}_{i=1}^N$  são as variáveis de folga,  $\mathbf{w}$  é um vetor normal e  $b$  é o bias. Reformulando (1), a função de Lagrange pode ser escrita como

$$L(\mathbf{W}, b, \xi, \alpha) = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^N \xi_i^2 - \sum_{i=1}^N \alpha_i (y_i(\mathbf{x}_i \mathbf{w} + b) - 1 + \xi_i) \quad (2)$$

no qual  $\{\alpha_i\}_{i=1}^N$  são os multiplicadores associados à restrição de igualdade do problema primal. Depois da obtenção das condições de otimalidade, o sistema KKT pode ser disposto como um sistema linear  $\mathbf{A}\mathbf{z} = \mathbf{b}$ , tal qual

$$\begin{bmatrix} 0 & -y^\top \\ y & \Omega + \gamma^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \quad (3)$$

em que  $\Omega \in \mathbb{R}^{N \times N}$  é uma matriz quadrada, que pode ser descrita como  $\Omega_{i,j} = y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$ , para  $i, j = 1, \dots, N$ . Ademais,  $y$  é um vetor linha que contém os rótulos de uma base e  $\mathbf{1}$  é um vetor de dimensão  $N$ . Para solucionar os sistema linear descrito, pode-se utilizar a inversão direta da matriz  $\mathbf{A}^{-1}$ , para achar  $\mathbf{z} = \mathbf{A}^{-1} \mathbf{b}$ . Depois de encontrar os multiplicadores de Lagrange e o bias, a saída pode ser calculada de acordo com a função

$$f(\mathbf{x}) = \text{sign} \left( \sum_i \alpha_i y_i \mathbf{x}^\top \mathbf{x}_i + b \right) \quad (4)$$

O produto  $\mathbf{x}^\top \mathbf{x}_i$  pode ser substituído por uma função de kernel da forma  $k(\mathbf{x}, \mathbf{x}_i)$ .

## 3. CLASSIFICADORES IP-LSSVM E P-LSSVM

Na literatura, existem alguns métodos que entregam um conjunto reduzido de vetores-suporte para o LSSVM, dois deles serão aplicados para comparação com o proposto nesse artigo. Esses foram IP-LSSVM (Carvalho and Braga, 2009) e *Pruning LSSVM* (P-LSSVM) (Suykens et al., 2000). Ambos, baseiam-se nos multiplicadores de Lagrange para a redução dos vetores-suporte. O IP-LSSVM tem três passos principais: primeiramente resolve-se o sistema linear  $\mathbf{A}\mathbf{z} = \mathbf{b}$  a partir da inversa, após isso os multiplicadores de Lagrange são ordenados e são retiradas as colunas da matriz  $\mathbf{A}$ , associadas aos menores valores. Por fim, resolve-se novamente o sistema, usando a pseudo inversa (5). O outro método P-LSSVM é semelhante. Os multiplicadores de Lagrange são ordenados, porém, desta vez, de acordo com o valor absoluto  $|\alpha_i|$  e são zerados gradativamente sem modificar a matriz  $\mathbf{A}$  até um critério de parada ser alcançado. Para ser possível uma comparação com o classificador proposto, o critério de parada aplicado ao P-LSSVM foi o percentual de redução.

$$\mathbf{z} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} \quad (5)$$

## 4. MÉTODO DE LEVENBERG-MARQUARDT

O método de Levenberg-Marquardt (LM) (Ranganathan, 2004) é um algoritmo de otimização iterativo que aplica treinamento em lote. Esse consiste em um aperfeiçoamento do método de Gauss-Newton. Considerando que o método de Gauss-Newton é uma variante do método de Newton. Logo, esse também utiliza a informação da derivada parcial de segunda ordem. Além da hessiana, o método de Levenberg-Marquardt também aplica a informação do gradiente.

$$L(\mathbf{z}) = \frac{1}{2} \sum_{i=1}^N e_i^2(\mathbf{z}) = \frac{1}{2} \mathbf{e}(\mathbf{z})^\top \mathbf{e}(\mathbf{z}) \quad (6)$$

Dado uma função de custo  $L(\mathbf{z})$  com base nos erros quadráticos, Equação (6), em que  $e_i(\mathbf{z})$  caracteriza o erro da  $i$ -ésima amostra e  $\mathbf{e}(\mathbf{z})$  é um vetor de erros. Aplicando uma expansão de Taylor em torno de um ponto  $\mathbf{z}_0$  em  $\nabla L(\mathbf{z})$  tem-se

$$\nabla L(\mathbf{z}) = \nabla L(\mathbf{z}_0) + (\mathbf{z} - \mathbf{z}_0)^\top \nabla^2 L(\mathbf{z}_0) + \dots \quad (7)$$

Sabendo que  $L(\mathbf{z})$  é convexa e duas vezes diferenciável, uma condição necessária e suficiente para  $\mathbf{z}$  ser um ponto ótimo é

$$\nabla L(\mathbf{z}) = 0 \quad (8)$$

Considerando apenas os termos de segunda ordem e desprezando o resto, na expansão de Taylor, Equação (7), pode-se aproximar  $\Delta \mathbf{z} = (\mathbf{z} - \mathbf{z}_0)$  de forma

$$\Delta \mathbf{z} = -[\nabla^2 L(\mathbf{z}_0)]^{-1} \nabla L(\mathbf{z}_0) \quad (9)$$

O gradiente  $\nabla L(\mathbf{z})$  e a hessiana  $\nabla^2 L(\mathbf{z})$  podem ser escritos em termos da matriz jacobiana  $\mathbf{J}(\mathbf{z})$ , então:

$$\nabla L(\mathbf{z}) = \mathbf{J}(\mathbf{z})\mathbf{e}(\mathbf{z}) \quad (10)$$

$$\nabla^2 L(\mathbf{z}) = \mathbf{J}^\top(\mathbf{z})\mathbf{J}(\mathbf{z}) + \sum_{i=1}^N e_i(\mathbf{z})\nabla^2 e_i(\mathbf{z}) \quad (11)$$

Levando em conta que o método Levenberg-Marquardt aplica conceitos do método de Gauss-Newton assume-se que:

$$\sum_{i=1}^N e_i(\mathbf{z})\nabla^2 e_i(\mathbf{z}) \approx 0 \quad (12)$$

Com isso o termo adicional em (12), pode ser retirado do cálculo da hessiana aproximada. A regra de atualização agora é dada por

$$\Delta \mathbf{z} = -[\mathbf{J}^\top(\mathbf{z})\mathbf{J}(\mathbf{z})]^{-1}\mathbf{J}(\mathbf{z})\mathbf{e}(\mathbf{z}) \quad (13)$$

Existe um problema na Equação (13), pois há possibilidade da matriz hessiana aproximada ser singular. Para solucionar isso, foi proposto por Levenberg (1944) a adição do termo  $\mu I$  ao cálculo desta matriz, no qual  $\mu$  é um escalar e  $I$  é a matriz identidade. Contudo, ainda existe um problema: quando  $\mu$  vai crescendo em algum momento a informação da matriz hessiana aproximada não será mais útil no cálculo da regra de atualização. Para esse problema, Marquardt (1963) propôs utilizar a diagonal da matriz hessiana aproximada ao invés da matriz identidade. Com essas alterações e sabendo que  $\mathbf{J}(\mathbf{z}) = -\mathbf{A}$  a regra de atualização será dada por

$$\mathbf{z} = \mathbf{z}_t + [\mathbf{A}^\top \mathbf{A} + \mu \text{diag}(\mathbf{A}^\top \mathbf{A})]^{-1} \mathbf{A}^\top \mathbf{e}(\mathbf{z}) \quad (14)$$

## 5. PROPOSTA: FSLM-LSSVM

A perda de esparsidade do LSSVM é um desafio importante a ser solucionado, uma vez que prejudica a performance do modelo em problemas com alta quantidade de dados. Esse obstáculo surge pela mudança nas restrições do problema primal do SVM, que passam de desigualdades para igualdades, afetando diretamente as derivações do sistema KKT, fazendo com que os multiplicadores de Lagrange associados sejam quase sempre não nulos. Para a obtenção de um modelo esparso, propõe-se nesse artigo uma solução iterativa com poda para o LSSVM, doravante denominado FSLM-LSSVM.

O classificador FSLM-LSSVM é um método derivado do LSSVM tradicional que visa a encontrar o ponto ótimo  $\mathbf{z}^*$  do sistema linear KKT de forma iterativa, ao mesmo tempo em que é removido uma quantidade fixa de vetores-suporte. Com esse propósito, foi adaptado o método de Levenberg-Marquardt para fins de remoção de vetores-suporte com base nos valores dos multiplicadores de Lagrange obtidos durante o processo iterativo para alcançar a solução ótima.

A análise dos multiplicadores de Lagrange pelo FSLM-LSSVM é feita a partir do seu valor absoluto (magnitude), similarmente ao que ocorre com o P-LSSVM. Uma análise detalhada para obtenção de conjuntos reduzidos em LSSVM pode ser vista em (ROCHA-NETO, 2011).

O algoritmo a seguir mostra o passo a passo do classificador FSLM-LSSVM. Além disso, foi feito um fluxograma (figura 1) com objetivo de demonstrar a lógica principal do algoritmo proposto. As entradas do algoritmo são: Matriz  $\mathbf{A}$ , vetor  $\mathbf{z}$ , vetor  $\mathbf{b}$ , coeficiente  $\mu$ , percentual de redução  $R$ , número de iterações  $N$  e faixa de remoção  $\rho$ . O FSLM-LSSVM realiza  $N+1$  iterações. Em cada uma é calculado um ajuste  $\Delta \mathbf{z}$  e, a partir desses, é verificado se o erro quadrático médio aumenta. Se sim, o vetor não será atualizado e o coeficiente  $\mu$  é reduzido. Caso contrário, o vetor é atualizado. Após esse passo, serão ordenados apenas os multiplicadores de Lagrange e, a partir desses, é verificado quais colunas de  $\mathbf{A}$  e quais linhas de  $\mathbf{z}$  são removidas. A remoção de uma coluna de  $\mathbf{A}$  é equivalente a zerar um multiplicador de Lagrange (Carvalho and Braga, 2009).

Considerando ainda que o método de Levenberg-Marquardt é iterativo, é interessante definir uma faixa de remoção (poda) de vetores-suporte durante a execução deste método. Por exemplo, tendo iterações na seguinte faixa  $[1, \dots, N]$ , as podas podem ocorrer nas seguintes iterações  $[\rho, \dots, N - \rho]$ . Vale destacar que nas simulações computacionais realizadas neste trabalho o valor de  $\rho = 3$ .

### Algoritmo: FSLM-LSSVM

**Entrada:**  $\mathbf{A}$ ,  $\mathbf{z}$ ,  $\mathbf{b}$ ,  $\mu$ ,  $R$ ,  $N$ ,  $\rho=3$

**Saída:**  $\mathbf{z}$

**início**

$M = \mathbf{A}.\text{colunas}$

$i = 1$

$K = R * M$

**repita**

$\mathbf{e}_i(\mathbf{z}_i) = \mathbf{b} - \mathbf{A}^\top \mathbf{z}_i$

$\Delta \mathbf{z} = [\mathbf{A}^\top \mathbf{A} + \mu \text{diag}(\mathbf{A}^\top \mathbf{A})]^{-1} \mathbf{A}^\top \mathbf{e}_i(\mathbf{z}_i)$

**se**  $\frac{\mathbf{e}_i(\mathbf{z}_i)^\top \mathbf{e}_i(\mathbf{z}_i)}{N} > \frac{\mathbf{e}_i(\mathbf{z}_i + \Delta \mathbf{z})^\top \mathbf{e}_i(\mathbf{z}_i + \Delta \mathbf{z})}{N}$  **então**

$\mu = \frac{1}{10} \mu$

**senão**

$\mathbf{z}_{i+1} = \mathbf{z}_i + \Delta \mathbf{z}$

**fim**

**se**  $\rho \leq i \leq N - \rho$  **então**

Remover  $\frac{K}{N-2\rho}$  colunas de  $\mathbf{A}$  e linhas de  $\mathbf{z}$  cujos valores de  $|\alpha_i|$  sejam os menores dentre os existentes.

$i = i + 1$

**até**  $i > N+1$ ;

**fim**

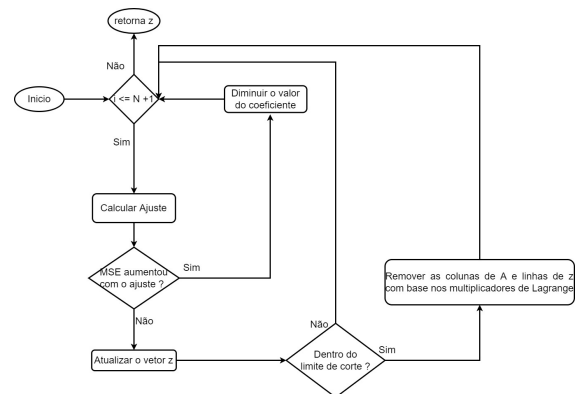


Figura 1. Fluxograma do algoritmo FSLM-LSSVM.

## 6. SIMULAÇÕES COMPUTACIONAIS

Tabela 1. Lista de bases de dados.

Base de dados	Abreviação	# Padrões	# Att
Ripley	RIP	1250	2
Haberman	HAB	306	3
Pima Indians Diabets	PID	768	8
Breast Cancer Winconsin	BCW	570	10
Vertebral Column Pathologies	VCP	310	6

Neste trabalho foram analisadas cinco bases de dados do mundo real. Essas foram adquiridas do repositório UCI Machine Learning UCI (2021). Os respectivos nomes, abreviações, quantidade de padrões (# Padrões) e número de atributos (# Att) estão disponíveis na Tabela 1.

Nos métodos avaliados, é necessário a definição de alguns hiperparâmetros, como o  $\gamma$  do sistema linear, o  $\mu$  da regra de decisão do algoritmo de Levenberg-Marquardt e a constante  $C$  do SVM. Os aplicados a essa pesquisa foram  $\mu = 0,0001$ ,  $\gamma$  variando entre 0,04 e 0,05 e o  $C$  entre 0 e 1. Foram feitas 30 iterações,  $N=30$ , do algoritmo de Levenberg-Marquardt com  $\rho = 3$ .

As implementações foram feitas em linguagem python e seus testes computacionais seguiram duas metodologias. Essas foram necessárias, para seguir a proposição original dos classificadores IP-LSSVM e P-LSSVM. Na primeira comparou-se o método FSLM-LSSVM com o IP-LSSVM. Já na segunda é utilizado o P-LSSVM. Em ambas as metodologias, para cada algoritmo foram feitas 30 realizações (rodadas completas) e retirou-se a média da taxa de acerto.

### 6.1 Metodologia 80-20

No primeiro momento, foram realizadas simulações computacionais com hold-out 80-20, no qual consiste em selecionar de forma aleatória 80 % dos dados para o treinamento do modelo e 20 % para testá-lo. Essa divisão foi feita para seguir a metodologia utilizada na proposição original do IP-LSSVM. Com essa, foi possível comparar o método FSLM-LSSVM com o IP-LSSVM. Na Tabela 2 estão algumas informações: a Acurácia, à direita, com o desvio padrão associado, à esquerda (Acc %), o número total de vetores-suporte do conjunto de treinamento (# $TS$ ), o número de vetores-suporte após a redução (# $VS$ ) e a porcentagem de redução aplicada (Red %).

A partir das tabelas 2 e 3, é possível avaliar o método proposto. Os resultados deste foram similares ao IP-LSSVM em alguns problemas. Já em outros, o FSLM-LSSVM foi superior. Ademais, mesmo com uma alta taxa de redução nos vetores-suporte, o modelo FSLM-LSSVM conseguiu se equiparar ao LSSVM padrão em quatro das cinco bases.

As Figuras 2 a 6 exibem os gráficos da acurácia (%) por tamanho da poda (%), para os classificadores FSLM-LSSVM e IP-LSSVM. O tamanho da poda indica quantos vetores-suporte foram removidos do conjunto do treinamento, variando de 10% a 90%.

A partir dos gráficos, é reafirmado que as taxas do FSLM-LSSVM ficaram muito próximas do IP-LSSVM, todavia o decaimento na taxa de acerto de acordo com a redução

foi mais rápida no IP-LSSVM em três das cinco bases testadas.

Tabela 2. Resultados com metodologia 80-20 aplicando kernel linear.

Base	Modelo	Acc (%)	# $TS$	# $VS$	Red (%)
VCP	SVM	85 ± 3,4	247	132	-
VCP	LSSVM	83,63 ± 4,5	247	247	-
VCP	IP - LSSVM <sub>20</sub>	72,98 ± 5,1	247	198	20
VCP	FSLM - LSSVM <sub>20</sub>	82,42 ± 5,2	247	198	20
VCP	IP - LSSVM <sub>50</sub>	72,66 ± 6,1	247	123	50
VCP	FSLM - LSSVM <sub>50</sub>	79,92 ± 3,7	247	123	50
VCP	IP - LSSVM <sub>80</sub>	72,10 ± 6,0	247	49	80
VCP	FSLM - LSSVM <sub>80</sub>	75,16 ± 4,7	247	49	80
HAB	SVM	73,79 ± 4,1	244	60	-
HAB	LSSVM	73,63 ± 3,5	244	244	-
HAB	IP - LSSVM <sub>20</sub>	69,03 ± 7,8	244	195	20
HAB	FSLM - LSSVM <sub>20</sub>	73,87 ± 5,6	244	195	20
HAB	IP - LSSVM <sub>50</sub>	63,79 ± 5,8	244	122	50
HAB	FSLM - LSSVM <sub>50</sub>	75,24 ± 3,8	244	122	50
HAB	IP - LSSVM <sub>80</sub>	63,23 ± 9,4	244	49	80
HAB	FSLM - LSSVM <sub>80</sub>	73,63 ± 4,4	244	49	80
BCW	SVM	97,59 ± 0,7	546	60	-
BCW	LSSVM	96,24 ± 1,1	546	546	-
BCW	IP - LSSVM <sub>20</sub>	97,45 ± 1,2	546	437	20
BCW	FSLM - LSSVM <sub>20</sub>	96,40 ± 1,3	546	437	20
BCW	IP - LSSVM <sub>50</sub>	97,12 ± 1,2	546	273	50
BCW	FSLM - LSSVM <sub>50</sub>	96,74 ± 1,4	546	273	50
BCW	IP - LSSVM <sub>80</sub>	96,64 ± 1,4	546	109	80
BCW	FSLM - LSSVM <sub>80</sub>	96,62 ± 1,6	546	109	80
PID	SVM	77,35 ± 3,0	614	332	-
PID	LSSVM	76,55 ± 2,5	614	614	-
PID	IP - LSSVM <sub>20</sub>	70,97 ± 3,9	614	491	20
PID	FSLM - LSSVM <sub>20</sub>	75,45 ± 3,2	614	491	20
PID	IP - LSSVM <sub>50</sub>	72,79 ± 3,2	614	307	50
PID	FSLM - LSSVM <sub>50</sub>	75,88 ± 2,8	614	307	50
PID	IP - LSSVM <sub>80</sub>	68,02 ± 3,0	614	123	80
PID	FSLM - LSSVM <sub>80</sub>	74,06 ± 4,2	614	123	80
RIP	SVM	88,0 ± 1,9	998	333	-
RIP	LSSVM	87,52 ± 1,1	998	998	-
RIP	IP - LSSVM <sub>20</sub>	87,76 ± 2,5	998	798	20
RIP	FSLM - LSSVM <sub>20</sub>	88,84 ± 1,5	998	798	20
RIP	IP - LSSVM <sub>50</sub>	87,84 ± 2,1	998	499	50
RIP	FSLM - LSSVM <sub>50</sub>	88,90 ± 1,4	998	499	50
RIP	IP - LSSVM <sub>80</sub>	88,20 ± 1,5	998	200	80
RIP	FSLM - LSSVM <sub>80</sub>	88,10 ± 1,6	998	200	80

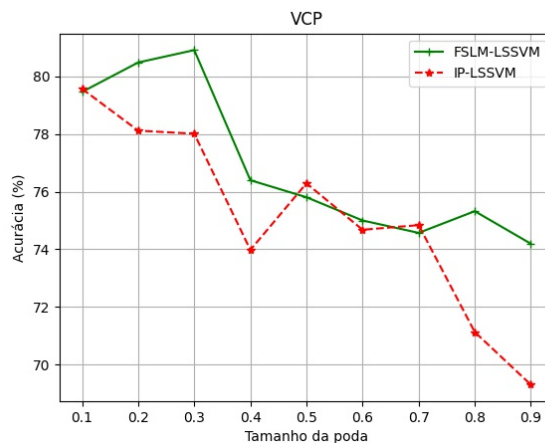


Figura 2. Acurácia versus tamanho da poda para VCP, com a metodologia 80-20.

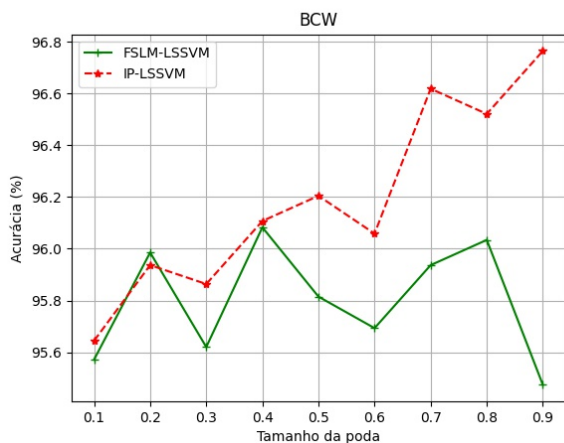


Figura 3. Acurácia *versus* tamanho da poda para BCW, com a metodologia 80-20.

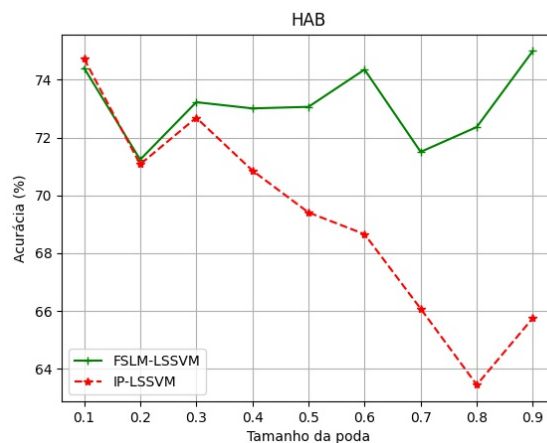


Figura 6. Acurácia *versus* tamanho da poda para HAB, com a metodologia 80-20.

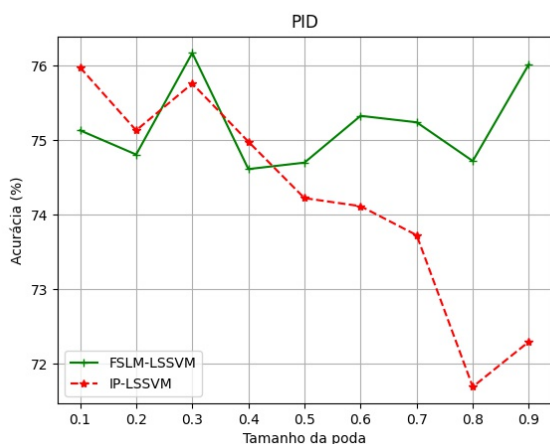


Figura 4. Acurácia *versus* tamanho da poda para PID, com a metodologia 80-20.

### 6.2 Metodologia 60-20-20

Na segunda parte da pesquisa, as simulações foram feitas com uma nova divisão, onde foi selecionada, de forma aleatória, 60 % dos dados para o treinamento, 20 % para conjunto de validação e 20 % para o conjunto de teste. Do mesmo modo, essa divisão foi feita para seguir a metodologia utilizada pelo classificador P-LSSVM. Com essa metodologia, pode-se comparar o método FSLM-LSSVM com o P-LSSVM. A Tabela 3 é similar à Tabela 2. Essa mostra os resultados com a metodologia 60-20-20. De acordo com os dados, nota-se que, mesmo com a redução no tamanho do conjunto de treinamento, o FSLM-LSSVM obteve resultados ótimos, equiparando-se ao LSSVM padrão. Além disso, os resultados dos modelos FSLM-LSSVM e P-LSSVM foram similares.

Nos gráficos das Figuras 7 a 11 está a acurácia (%) pelo tamanho da poda (%), para os classificadores FSLM-LSSVM e P-LSSVM. Analisando-os, é possível perceber o mesmo comportamento da metodologia anterior, em que os resultados foram próximos. Todavia o decaimento na taxa de acerto é maior no P-LSSVM em algumas bases.

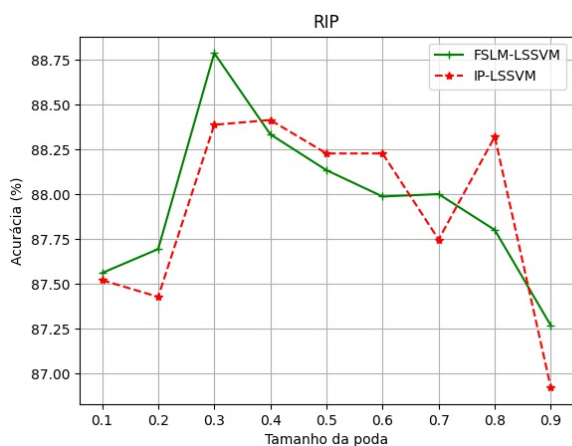


Figura 5. Acurácia *versus* tamanho da poda para RIP, com a metodologia 80-20.

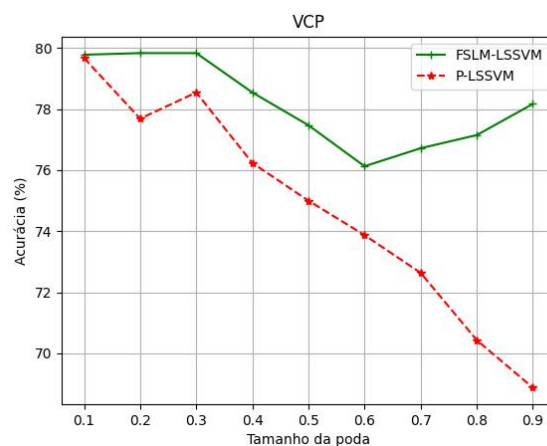


Figura 7. Acurácia *versus* tamanho da poda para VCP, com a metodologia 60-20-20.

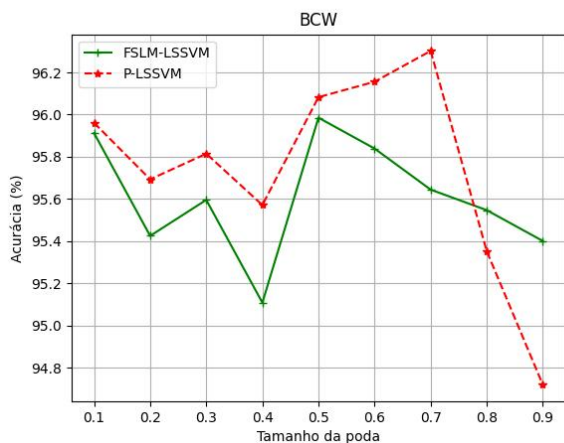


Figura 8. Acurácia *versus* tamanho da poda para BCW, com a metodologia 60-20-20.

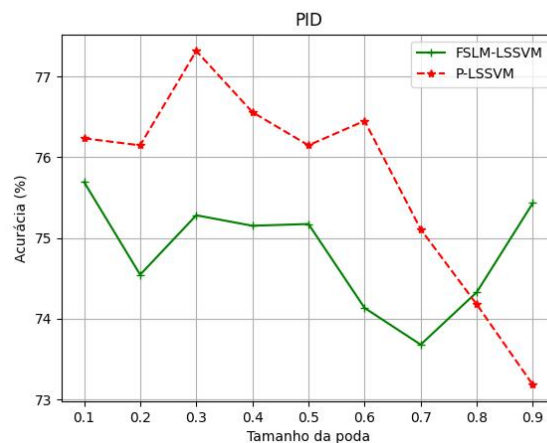


Figura 9. Acurácia *versus* tamanho da poda para PID, com a metodologia 60-20-20.

Tabela 3. Resultados com metodologia 60-20-20 aplicando kernel linear.

Base	Modelo	Acc (%)	#TS	#VS	Red (%)
VCP	SVM	86,02 ± 3,8	186	124	-
VCP	LSSVM	83,60 ± 3,4	186	186	-
VCP	<i>P</i> - LSSVM <sub>20</sub>	79,76 ± 5,7	186	149	20
VCP	<i>FSLM</i> - LSSVM <sub>20</sub>	81,51 ± 4,7	186	149	20
VCP	<i>P</i> - LSSVM <sub>50</sub>	79,35 ± 4,8	186	93	50
VCP	<i>FSLM</i> - LSSVM <sub>50</sub>	76,34 ± 6,0	186	93	50
VCP	<i>P</i> - LSSVM <sub>80</sub>	75,73 ± 8,0	186	37	80
VCP	<i>FSLM</i> - LSSVM <sub>80</sub>	75,65 ± 7,2	186	37	80
HAB	SVM	74,73 ± 1,7	184	128	-
HAB	LSSVM	75,05 ± 3,4	184	184	-
HAB	<i>P</i> - LSSVM <sub>20</sub>	74,57 ± 3,6	184	147	20
HAB	<i>FSLM</i> - LSSVM <sub>20</sub>	74,73 ± 6,1	184	147	20
HAB	<i>P</i> - LSSVM <sub>50</sub>	73,71 ± 5,9	184	92	50
HAB	<i>FSLM</i> - LSSVM <sub>50</sub>	73,71 ± 5,5	184	92	50
HAB	<i>P</i> - LSSVM <sub>80</sub>	71,94 ± 4,6	184	37	80
HAB	<i>FSLM</i> - LSSVM <sub>80</sub>	73,71 ± 4,5	184	37	80
BCW	SVM	97,54 ± 1,0	410	49	-
BCW	LSSVM	96,11 ± 1,6	410	410	-
BCW	<i>P</i> - LSSVM <sub>20</sub>	96,18 ± 1,6	410	328	20
BCW	<i>FSLM</i> - LSSVM <sub>20</sub>	97,40 ± 1,3	410	328	20
BCW	<i>P</i> - LSSVM <sub>50</sub>	95,99 ± 1,3	410	205	50
BCW	<i>FSLM</i> - LSSVM <sub>50</sub>	96,74 ± 1,2	410	205	50
BCW	<i>P</i> - LSSVM <sub>80</sub>	95,62 ± 1,6	410	82	80
BCW	<i>FSLM</i> - LSSVM <sub>80</sub>	97,01 ± 1,0	410	82	80
PID	SVM	76,95 ± 2,3	461	312	-
PID	LSSVM	77,45 ± 2,0	461	461	-
PID	<i>P</i> - LSSVM <sub>20</sub>	76,41 ± 3,6	461	369	20
PID	<i>FSLM</i> - LSSVM <sub>20</sub>	76,69 ± 3,2	461	369	20
PID	<i>P</i> - LSSVM <sub>50</sub>	76,41 ± 3,1	461	230	50
PID	<i>FSLM</i> - LSSVM <sub>50</sub>	75,48 ± 3,5	461	230	50
PID	<i>P</i> - LSSVM <sub>80</sub>	75,74 ± 2,4	461	92	80
PID	<i>FSLM</i> - LSSVM <sub>80</sub>	74,20 ± 3,5	461	92	80
RIP	SVM	88,21 ± 1,5	750	315	-
RIP	LSSVM	88,33 ± 1,4	750	750	-
RIP	<i>P</i> - LSSVM <sub>20</sub>	88,01 ± 2,0	750	600	20
RIP	<i>FSLM</i> - LSSVM <sub>20</sub>	88,02 ± 1,6	750	600	20
RIP	<i>P</i> - LSSVM <sub>50</sub>	88,35 ± 1,7	750	375	50
RIP	<i>FSLM</i> - LSSVM <sub>50</sub>	88,32 ± 1,8	750	375	50
RIP	<i>P</i> - LSSVM <sub>80</sub>	87,84 ± 1,8	750	150	80
RIP	<i>FSLM</i> - LSSVM <sub>80</sub>	87,85 ± 1,6	750	150	80

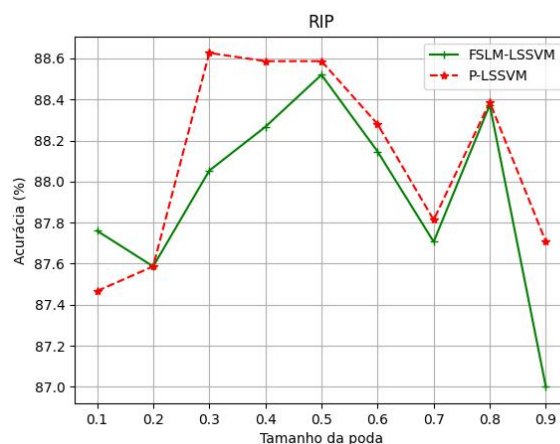


Figura 10. Acurácia *versus* tamanho da poda para RIP, com a metodologia 60-20-20.

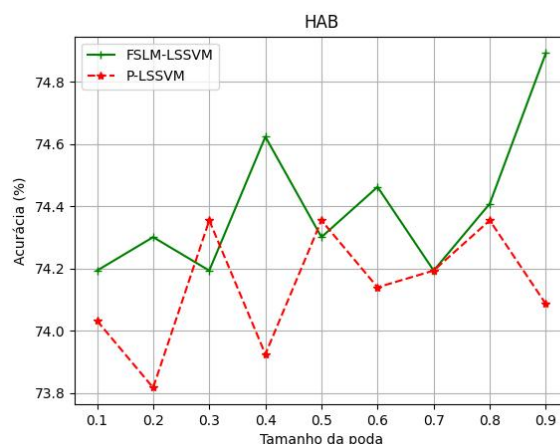


Figura 11. Acurácia *versus* tamanho da poda para HAB, com a metodologia 60-20-20.

## 7. CONCLUSÕES

Neste artigo, foi proposto o classificador FSLM-LSSVM, o qual foi obtido a partir do método de Levenberg-Marquardt com poda iterativa. Notou-se, de maneira geral, que o método obteve um bom desempenho, apresentando-se como uma alternativa eficiente para resolver o problema da perda de esparsidade do LSSVM. Além disso, mesmo com alta taxa de redução o FSLM-LSSVM se mostrou eficiente, superando os classificadores IP-LSSVM e P-LSSVM em alguns problemas. Para trabalhos futuros, pretende-se comparar o método proposto com outros mais novos da literatura.

## REFERÊNCIAS

- Burges, C. (1996). Simplified support vector decision rules. 71–77.
- Carvalho, B. and Braga, A. (2009). IP-LSSVM: A two-step sparse classifier. *Pattern Recognition Letters*, 30(16), 1507–1515. doi:<https://doi.org/10.1016/j.patrec.2009.07.022>. URL <https://www.sciencedirect.com/science/article/pii/S0167865509002086>.
- Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley & Sons, New York, NY, USA, second edition.
- Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2), 164–168. URL <http://www.jstor.org/stable/43633451>.
- Marquardt, D.W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431–441. URL <http://www.jstor.org/stable/2098941>.
- Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer, New York, NY, USA, 2e edition.
- Ranganathan, A. (2004). The levenberg-marquardt algorithm. *Academia.edu*.
- ROCHA-NETO, A.R. (2011). *Sinpatco II: novas estratégias de aprendizado de máquina para classificação de patologias da coluna vertebral*. Ph.D. thesis, Universidade Federal do Ceará.
- Suykens, J. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293–300. doi:10.1023/a:1018628609742. URL <https://doi.org/10.1023/a:1018628609742>.
- Suykens, J., Lukas, L., and Vandewalle, J. (2000). Sparse least squares support vector machine classifiers. volume 1, 37–42.
- UCI (2021). Uci machine learning. URL <https://archive.ics.uci.edu/ml/index.php>.