UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

DOUTORADO EM ENGENHARIA DE TELEINFORMÁTICA

FELIPE PINTO MARINHO

MÉTODOS PARA TREINAMENTO RÁPIDO E ESPARSO DE MÁQUINAS DE VETORES-SUPORTE DE MÍNIMOS QUADRADOS - UMA ABORDAGEM DUAL

FORTALEZA

2025

FELIPE PINTO MARINHO

MÉTODOS PARA TREINAMENTO RÁPIDO E ESPARSO DE MÁQUINAS DE VETORES-SUPORTE DE MÍNIMOS QUADRADOS - UMA ABORDAGEM DUAL

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Orientador: Prof. Dr. Ajalmar Rêgo da Rocha Neto.
Coorientador: Prof. Dr. Paulo Alexandre Costa Rocha.

FORTALEZA

2025

FELIPE PINTO MARINHO

MÉTODOS PARA TREINAMENTO RÁPIDO E ESPARSO DE MÁQUINAS DE VETORES-SUPORTE DE MÍNIMOS QUADRADOS - UMA ABORDAGEM DUAL

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Aprovada em: 19 de Dezembro de 2025.

BANCA EXAMINADORA

Prof. Dr. Ajalmar Rêgo da Rocha Neto (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Paulo Alexandre Costa Rocha (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Paulo Cesar Cortez
Universidade Federal do Ceará (UFC)

Prof. Dr. Rodrigo de Melo Souza Veras
Universidade Federal do Piauí (UFPI)

Prof. Dr. Auzuir Ripardo de Alexandria
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

À minha família, em particular à minha mãe e à minha esposa pelo convívio, suporte e paciência durante toda a caminhada desde a graduação até esta última etapa de doutorado.

AGRADECIMENTOS

Agradeço a Deus, por todos os dias me conceder força e saúde para seguir pela caminhada acadêmica que por muitas vezes é árdua e difícil.

Ao Prof. Dr. Ajalmar Rêgo da Rocha Neto, pela excelente orientação, confiança e paciência com o trabalho desenvolvido.

Ao meu coorientador, Prof. Dr. Paulo Alexandre Costa Rocha, por ter me acolhido com paciência e confiança ainda nos tempos de graduação e mestrado. Pelo exemplo de profissional, tanto no âmbito do ensino como na pesquisa, que inspira e contagia a todos pelas conversas e sugestões sempre perspicazes.

Aos professores participantes da banca examinadora pelo tempo, pelas valiosas colaborações e sugestões.

Aos colegas de pesquisa, pelo suporte no desenvolvimento de diversos artigos, em especial, a Victor Oliveira Santos, Francisco Diego Vidal Bezerra e a Wellington Dantas de Almeida pelos vários *insights*, revisões, melhorias na escrita, etc.

Aos colegas da turma de doutorado, pelas reflexões, críticas e sugestões recebidas.

À minha esposa, Letícia Rayanne, por todo amor, carinho, paciência e apoio concedido durante todo o doutorado que foi de grande valia para a conclusão desta etapa.

À minha mãe, por toda a força, incentivo e luta para que eu pudesse seguir pelo caminho acadêmico sempre com foco apenas nos estudos, além do incentivo concedido nos momentos de desânimos.

Aos colegas do Banco do Nordeste, em especial a Polycarpo Neto, Isac Lira, Felype Bastos e Edyvalberty Alenquer e também aos colegas da Dhauz, por todos os ensinamentos práticos relacionados aos temas de aprendizado de máquina e ciência de dados que muito auxiliaram no desenvolvimento das implementações numéricas que culminaram na obtenção dos resultados aqui apresentados.

Por fim, agradeço à Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) por todo o suporte financeiro fornecido na forma de bolsa de pesquisa.

"Os que se encantam com a prática sem a ciência são como os timoneiros que entram no navio sem timão nem bússola, nunca tendo certeza do seu destino" (Leonardo da Vinci)

RESUMO

O modelo de máquinas de vetores-suporte de mínimos quadrados é uma variante do clássico modelo de máquinas de vetores-suporte que utiliza restrições de igualdade na formulação de seu problema primal. Isto permite a obtenção de um sistema linear ao aplicar as condições de Karush-Kuhn-Tucker para a otimalidade do problema, simplificando consideravelmente o treinamento deste modelo quando comparado ao ajuste das máquinas de vetores-suporte. No entanto, uma desvantagem dessa formulação está no fato de que o vetor ótimo de multiplicadores de Lagrange do problema é não esparso. Assim, todos os padrões de treinamento serão considerados vetores-suporte, fazendo com que o estágio de predição seja oneroso quando se trabalha com grandes bases de dados. Em muitos casos, a solução do sistema é dada pelo uso de métodos iterativos baseados em direções conjugadas, o que por um lado é vantajoso, uma vez que evita as dificuldades numéricas relacionadas a inversão de matrizes, por outro, torna o estágio de treinamento lento para bases com alta volumetria, já que é necessário operar com matrizes de *kernel* densas. Neste contexto, propõem-se duas novas metodologias para o treinamento rápido e esparso de máquinas de vetores-suporte de mínimos quadrados. Na primeira abordagem, o problema dual das máquinas de vetores-suporte de mínimos quadrados é resolvido via algoritmo de otimização sequencial mínima com uma nova direção de descida conjugada de três termos, que combinada a uma estratégia de seleção de conjunto de trabalho baseada no ganho funcional permite uma aceleração na convergência, reduzindo o número de iterações quando comparado ao algoritmo de otimização sequencial mínima padrão. Além disso, um processo de poda iterativa baseado no ganho funcional do problema de otimização é adotado com o intuito de esparsificar os multiplicadores de Lagrange obtidos. Por fim, a última proposta consiste no uso de um novo método do gradiente conjugado espectral para a solução do problema dual correspondente e esparsificação via poda iterativa utilizando a proximidade do padrão ao hiperplano de decisão como critério para a remoção. Experimentos numéricos realizados sobre várias bases de dados reais e artificiais comprovam que as duas abordagens apresentam desempenho competitivo, com rápido treinamento e alto nível de esparsidade dos multiplicadores de Lagrange. Para as bases de classificação binária, o ganho de esparsidade chegou à aproximadamente 80% quando comparado ao total de amostras de treino para o conjunto de dados considerado. A redução no tempo de treinamento foi de cerca de 99.9% em relação às máquinas de vetores-suporte de mínimos quadrados padrão. Para bases de maior volumetria, as propostas foram as únicas que forneceram um tempo de treinamento hábil com estabilidade na convergência. A qualidade

das fronteiras de decisão foram ainda analisadas para conjuntos de dados sintéticos, em que os resultados indicam geração de fronteiras similares ao modelo de *benchmarking* considerado, ratificando a capacidade preditiva das novas metodologias. Por fim, os resultados para as bases de regressão indicam que a proposta baseada no gradiente conjugado espectral pode ser uma alternativa esparsa e com rápido treinamento ao modelo de regressores de vetores-suporte de mínimos quadrados.

Palavras-chave: máquinas de vetores-suporte de mínimos quadrados; otimização sequencial mínima; direções conjugadas; gradiente conjugado espectral, esparsidade.

ABSTRACT

The least squares support vector machine model is a variant of the classical support vector machine model that employs equality constraints in the formulation of its primal problem. This allows the derivation of a linear system when applying the Karush–Kuhn–Tucker optimality conditions, considerably simplifying the training of this model when compared to the adjustment of support vector machines. However, a drawback of this formulation lies in the fact that the optimal vector of Lagrange multipliers of the problem is dense. Thus, all training patterns are considered support vectors, making the prediction stage computationally expensive when working with large datasets. In many cases, the solution of the system is obtained through the use of iterative methods based on conjugate directions, which, on the one hand, is advantageous since it avoids numerical difficulties related to matrix inversion, but, on the other hand, makes the training stage slow for datasets with high volume, as it is necessary to operate with dense kernel matrices. In this context, two new methodologies are proposed for the fast and sparse training of least squares support vector machines. In the first approach, the dual problem of least squares support vector machines is solved via a sequential minimal optimization algorithm with a new three-term conjugate descent direction which, combined with a working set selection strategy based on functional gain, allows an acceleration in convergence, reducing the number of iterations when compared to the standard sequential minimal optimization algorithm. In addition, an iterative pruning process based on the functional gain of the optimization problem is adopted in order to sparsify the obtained Lagrange multipliers. Finally, the last proposal consists of the use of a new spectral conjugate gradient method for solving the corresponding dual problem and sparsification through iterative pruning using the proximity of the pattern to the decision hyperplane as the criterion for removal. Numerical experiments carried out on several real and artificial datasets demonstrate that both approaches present competitive performance, with fast training and a high level of sparsity of the Lagrange multipliers. For binary classification datasets, the sparsity gain reached approximately 80% when compared to the total number of training samples for the considered dataset. The reduction in training time was approximately 99.9% in relation to standard least squares support vector machines. For datasets with higher volume, the proposals were the only ones that provided feasible training time with stable convergence. The quality of the decision boundaries was further analyzed for synthetic datasets, where the results indicate the generation of boundaries similar to the considered benchmarking model, confirming the predictive capability of the new methodologies. Finally, the results for regression datasets

indicate that the proposal based on the spectral conjugate gradient method may be a sparse and fast-training alternative to the least squares support vector machine regression model.

Keywords: least squares support vector machines; sequential minimal optimization; conjugate directions; spectral conjugate gradient, sparsity.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

AdaGrad	*adaptive subgradient*
Adam	*adaptive moment estimation*
BFGS	Broyden-Fletcher-Goldfarb-Shanno
CG	*conjugate gradient*
CSMO	*conjugate functional gain SMO*
CSMO-LSSVM	*conjugate functional gain SMO LSSVM*
DNI	*direct normal irradiance*
EFB	*exclusive feature bundling*
FGWSS	*functional gain working selection strategy*
FSLM-LSSVM	*fixed sized Levenberg-Marquardt LSSVM*
GHI	*global horizontal irradiance*
GMDH	*group method handling of data*
GOSS	*gradient-based one side sampling*
IP-LSSVM	*importance pruning least squares support vector machine*
KKT	Karush-Kuhn-Tucker
LightGBM	*light gradient boosting machine*
LM	Levenberg-Marquardt
LSSVC	*least squares support vector classifier*
LSSVM	*least squares support vector machine*
LSSVR	*least squares support vector regressor*
MLP	*multilayer percepton*
MVP	*maximum violation pair*
NWP	*numerical weather prediction*
P-LSSVM	*prunning least squares support vector machine*
QP	*quadratic programming*
RFE	*recursive feature elimination*
RLARS	*revised least angle regression*
RMSProp	*root mean square propagation*
SCG	*spectral conjugate gradient*
SCG-LSSVM	*spectral conjugate gradient LSSVM*

SMO	*sequential minimal optimization*
SMW	Sherman-Morrison-Woodbury
SVC	*support vector classifier*
SVM	*support vector machine*
SVR	*support vector regressor*
TCSMO	*three-term conjugate-like descent SMO*
TCSMO-LSSVM	*three-term conjugate-like LSSVM*
UCI	*UCI machine learning repository*
XGBoost	*extreme gradient boosting*

LISTA DE SÍMBOLOS

$\boldsymbol{x}$	Vetor de atributo de entrada
$\boldsymbol{y}$	Vetor de atributo de saída ou alvo
$\boldsymbol{w}$	Vetor de parâmetros/pesos para o modelo
$\boldsymbol{\varepsilon}$	Vetor de erros
b	Termo de viés ou intercepto do modelo
$\phi(.)$	Mapeamento para um espaço de atributos de alta dimensão
$\mathbb{R}^n$	Espaço euclidiano n-dimensional
∞	Infinito
γ	Termo de regularização do LSSVM primal
$\mathscr{L}$	Lagrangeana de um problema de otimização
$\boldsymbol{\alpha}$	Vetor de multiplicadores de Lagrange
$\boldsymbol{K}$	Matriz de *kernel*
$\boldsymbol{I}$	Matriz identidade
$\mathscr{K}(,)$	Função de *kernel* de Mercer
$\boldsymbol{A}^{-1}$	Inversa da matriz $\boldsymbol{A}$
$\boldsymbol{A}^{\dagger}$	Pseudo-Inversa de Moore-Penrose da matriz $\boldsymbol{A}$
$\nabla(.)$	Gradiente de uma função
$\nabla^2(.)$	Hessiana de um função
$\boldsymbol{J}(.)$	Matriz Jacobiana de uma função
μ	Termo de regularização de Tikhonov
$\partial(.)$	Derivada parcial de uma função
$diag(.)$	Diagonal principal de uma matriz
N_{max}	Número máximo de iterações
κ	Termo de definição da janela de poda
t	Indicador de iteração
R	Porcentagem de redução

$\mathscr{D}(.)$	Função objetivo dual
$\tilde{\boldsymbol{K}}$	Matriz de *kernel* regularizada
$\boldsymbol{g}(.)$	Vetor gradiente em um ponto
$\Delta\boldsymbol{\alpha}$	Atualização da variável dual
$f_G(.)$	Ganho funcional em iterações sucessivas
ρ_k	Tamanho do passo de aprendizagem na k-ésima iteração
r_k	Parâmetro conjugado na k-ésima iteração para o método CFSMO-LSSVM
δ_k	Parâmetro conjugado na k-ésima iteração para o método TCSMO-LSSVM
μ_k	Parâmetro conjugado de três termos na k-ésima iteração
$\lambda_{min}(.)$	Menor autovalor de uma matriz
θ_k	Parâmetro conjugado na k-ésima iteração para o método SCG-LSSVM
β_k	Parâmetro espectral na k-ésima iteração para o método SCG-LSSVM
$\boldsymbol{B}_k$	Hessiana aproximada pelo esquema BFGS na k-ésima iteração
R^2	Coeficiente de determinação

SUMÁRIO

1 INTRODUÇÃO

Vários foram os avanços tecnológicos que proporcionaram grandes transformações na sociedade ao longo do tempo. A revolução agrícola (Grinin e Grinin, 2013), que permitiu a transição do nomadismo para sociedades agrícolas. A Revolução Industrial, que proporcionou a adoção de máquinas a vapor, produção em massa, ferrovias, interligação de fábricas e cidades inteiras, além de ter transformado os padrões de trabalho, urbanização, transporte, comércio e relações sociais (Jiang, 2024) e, mais recentemente, a revolução da informação com a emergência dos computadores, da internet, dos dispositivos móveis e da computação em nuvem (Hilbert, 2020).

O advento de soluções baseadas em inteligência artificial tem impactado consideravelmente o modelo de trabalho atual, induzindo o que tem sido chamado de quarta revolução industrial (Lund *et al.*, 2024). O que inicialmente, focou em modelos teóricos de máquinas com capacidade de realizar tarefas cognitivas (Turing, 2007), desenvolveu-se e tomou ares mais aplicados graças aos diversos ramos que compõem a área de inteligência artificial.

O aprendizado de máquina é um destes ramos e caracteriza-se por fornecer modelos que conseguem detectar padrões implícitos em uma massa significativa de dados. O processo de aprendizagem é crucial para uma boa generalização dos modelos, que hoje, são bem consolidados e largamente aplicados em diversas tarefas. Dentre estes, destacam-se as redes perceptron multicamadas (*multilayer percepton*, MLP) (Rosenblatt, 1958; Minsky e Papert, 1969; Rumelhart *et al.*, 1986), as máquinas de vetores-suporte (*support vector machine*, SVM) (Cortes e Vapnik, 1995) e as árvores de decisão (Breiman *et al.*, 2017).

As contribuições teóricas em novos modelos de aprendizado de máquina considerando ambos os paradigmas supervisionado e não supervisionado, têm permitido grandes avanços em áreas diversas como modelagem e controle de sistemas físicos (Haller e Kaszás, 2024; Brunton *et al.*, 2025), sistemas de recomendação (Da'u e Salim, 2020), análise de redes sociais (Park *et al.*, 2024), geração de áudio (Oord *et al.*, 2016), imagem (Shaham *et al.*, 2019) e vídeo (Ho *et al.*, 2022) com a denominada inteligência artificial generativa, etc. Em todas estas aplicações, fatores como alta dimensionalidade e ocorrências de não linearidades aumentam a complexidade envolvendo a modelagem do problema o que faz com que estes novos modelos necessitem de treinamento em grandes bases de dados para se ter uma capacidade preditiva adequada.

Além disso, dada a rápida evolução tecnológica na capacidade de processamento

de *hardware*, bem como, na alta disponibilidade de dados obtidos de fontes diversas, fazem com que aplicações que envolvam a extração de conhecimento de grandes bases de dados sejam bastantes comuns atualmente (Tsai *et al.*, 2015).

Também vale destacar, os muitos exemplos da necessidade de aprendizado em tempo real (*online*) por parte de modelos de aprendizado de máquina, principalmente em dispositivos embarcados com processamento computacional limitado: Dispositivos de predição climática de baixo custo (Marinho *et al.*, 2022; Marinho *et al.*, 2020; El-Amarty *et al.*, 2024; Marinho *et al.*, 2024), robótica aeroespacial (Hovell, 2022; Elkins, 2024; Hmede *et al.*, 2022a) e aplicações onde a tomada de decisão em tempo real é relevante como em veículos autônomos (Chen e Lv, 2022; Yan *et al.*, 2023).

Dado que modelos de aprendizado profundo (*deep learning*) necessitam estimar uma quantidade massiva de parâmetros e, portanto, demandam um processamento computacional oneroso, versões leves de clássicos modelos de aprendizado supervisionado são preferíveis em aplicações envolvendo aprendizado *online* (Florêncio *et al.*, 2020; Dias *et al.*, 2020; Dias *et al.*, 2018).

Essas versões leves de modelos de aprendizado supervisionado podem ser utilizados como uma camada de predição para redes neurais profundas em substituição a redes MLP densas, permitindo uma redução no número de parâmetros a ser aprendido e conduzindo a uma redução do custo da rede como um todo. Esta abordagem híbrida pode ser particularmente útil em cenários em que interpretabilidade, robustez e eficiência no treinamento são desejadas (Chen *et al.*, 2024; Mehrkanoon *et al.*, 2017).

Dentro desta categoria, destaca-se as SVMs, por apresentarem desempenho preditivo competitivo com garantias de minimização de risco estrutural e empírico, além de apresentar robusto formalismo matemático em sua concepção (Cortes e Vapnik, 1995). Esta metodologia apresenta versões específicas para tarefas de classificação e de aproximação de funções. No primeiro caso são denominadas classificadores de vetores-suporte (*support vector classifier*, SVC) e no segundo regressores de vetores-suporte (*support vector regressor*, SVR).

Máquinas de vetores-suporte de mínimos quadrados (*least squares support vector machine*, LSSVM) são uma variante das bem estabelecidas SVMs, que também traz consigo as mesmas vantagens desta última, com versões próprias para classificação e regressão chamados classificadores de vetores-suporte de mínimos quadrados (*least squares support vector classifier*, LSSVC) e regressores de vetores-suporte de mínimos quadrados (*least squares support vector*

regressor, LSSVR), respectivamente.

LSSVMs utilizam apenas restrições de igualdade no problema de maximização da margem, que ao aplicar as condições de Karush-Kuhn-Tucker (KKT) para a otimalidade acabam resultando na solução de um sistema linear KKT (Suykens e Vandewalle, 1999). Tal procedimento simplifica consideravelmente o treinamento do modelo quando comparado ao treinamento da SVM que precisa solucionar um programa quadrático (*quadratic programming*, QP) para o seu ajuste (Cristianini e Shawe-Taylor, 2000).

Apesar de ter um treinamento rápido, a LSSVM apresenta soluções não esparsas, fazendo com que a maioria dos padrões de treinamento sejam considerados vetores-suporte, impactando negativamente na fase de predição em cenários de processamento de grandes bases de dados e em aplicações de aprendizado *online*. Além disso, mesmo com a simplificação no treinamento, dependendo do tamanho da base de dados, o ajuste das LSSVMs pode ser ineficiente, já que para resolver o sistema KKT é necessário manipulações numéricas com uma matriz de *kernel* densa.

Nesse sentido, diversos pesquisadores propuseram algoritmos de treinamento rápido. Jiao *et al.* (2007a) utilizou um algoritmo aproximado para treinar rapidamente a LSSVM e melhorar sua esparsidade. Yang *et al.* (2010) propôs o uso de um algoritmo de poda para resolver eficientemente o problema de otimização da LSSVM. Li *et al.* (2013) desenvolveram um método iterativo rápido baseado em dados individuais para treinar LSSVMs irrestritas. Xia (2018) empregou fatoração QR para treinar LSSVMs esparsas. Chua (2003) propôs um método computacionalmente eficiente para resolver LSSVCs em larga escala baseado na técnica de inversão de matrizes de Sherman-Morrison-Woodbury (SMW). Entretanto, esse algoritmo possui altas exigências quanto ao mapeamento do *kernel*, sendo necessário especificar sua forma explícita.

No que diz respeito às metodologias que realizam a esparsificação do modelo LS-SVM, elas podem ser divididas em: (i) Métodos de redução e (ii) Métodos diretos. Os métodos de redução são aqueles que treinam a LSSVM em um conjunto de treinamento reduzido, onde existem padrões com maiores chances de serem vetores-suporte. Para a categoria de métodos diretos, a esparsidade é forçada desde o início, geralmente ainda durante a resolução do problema de otimização (Mall e Suykens, 2015).

Dentro da categoria de métodos de redução, destacam-se as seguintes variantes: as máquinas de vetores-suporte de mínimos quadrados com poda (*prunning least squares support*

vector machine, P-LSSVM) (Suykens *et al.*, 2000), o classificador de vetores-suporte esparso em duas etapas (*importance pruning least squares support vector machine*, IP-LSSVM) (Carvalho e Braga, 2009), o algoritmo genético mono e multi-objetivo para LSSVM esparso (Silva *et al.*, 2015) e a heurística *opposite maps* (Neto e Barreto, 2013). Para métodos diretos, destacam-se as metodologias: o esquema de aproximação esparsa rápida para LSSVM (Jiao *et al.*, 2007b), a decomposição de Cholesky pivotada da LSSVM primal (Zhou, 2015) e a LSSVM primal esparsa via regressão de menor ângulo revisada (*revised least angle regression*, RLARS) (Zhou e Liu, 2017).

Ainda considerando contribuições mais recentes na categoria de métodos de redução, destaca-se o trabalho desenvolvido em Leao e Neto (2022) que utiliza o algoritmo de Levenberg-Marquardt (LM) para a solução numérica do sistema KKT no treinamento da LSSVM. Para esparsificar a solução obtida, um esquema de poda iterativa é adotado, em que os padrões com menores valores absolutos de multiplicadores de Lagrange são removidos em cada iteração. O processo de poda é finalizado quando se alcança uma determinada quantidade de vetores-suporte especificada pelo usuário o que caracteriza tal procedimento como sendo de tamanho fixado.

Destaca-se ainda o recente uso de abordagens baseadas na solução do problema dual da LSSVM. Em López e Suykens (2011) o clássico algoritmo de otimização sequencial mínima (*sequential minimal optimization*, SMO) (Platt, 1998), utilizado para o rápido treinamento de modelos SVM, foi adaptado para a solução do problema dual da LSSVM, considerando a estratégia de primeira e segunda ordem na seleção do conjunto de trabalho.

Em Yu *et al.* (2023b), os autores propuseram o uso de uma direção de descida conjugada com estratégia de seleção do conjunto de trabalho baseado no ganho funcional (*functional gain working selection strategy*, FGWSS) permitindo maiores ganhos funcionais a cada iteração do que o SMO padrão, o modelo resultante foi chamado SMO conjugado de ganho funcional (*conjugate functional gain SMO*, CSMO).

Em Yu *et al.* (2023a), uma nova proposta para o treinamento rápido de modelos SVM foi apresentada, onde foi utilizada uma direção de descida conjugada de três termos (Beale, 1972), o que permitiu um maior ganho funcional do que ambos SMO padrão e CSMO.

Diante do exposto, propõe-se dois novos métodos para o treinamento rápido e esparso de modelos LSSVMs. Na primeira proposta, o problema dual para o treinamento da LSSVM é resolvido através do algoritmo SMO de três termos conjugados (*three-term conjugate-like descent SMO*, TCSMO), de maneira análoga à abordagem adotada em Yu *et al.* (2023a) para

SVMs, porém adaptado para o caso de LSSVMs. A nova direção de descida proposta para atualização dos multiplicadores de Lagrange permite um ganho funcional maior que a versão conjugada padrão descrita em Yu *et al.* (2023b) e, consequentemente, superior aos proporcionados pelos algoritmos SMO de primeira e segunda ordem. Como resultado, essa proposta oferece treinamento rápido, embora ainda gere soluções não esparsas para a LSSVM.

Para abordar o problema de esparsificação, emprega-se um procedimento de poda iterativa na solução obtida pelo TCSMO. A ideia é semelhante à utilizada em Zeng e Chen (2005), em que os padrões que contribuem menos para o ganho funcional do problema dual são removidos. A poda continua até que o número de amostras resultantes seja equivalente à porcentagem de redução estabelecida, caracterizando-se assim como um procedimento de tamanho fixo.

A segunda proposta ainda resolve o problema dual da LSSVM, mas agora empregando um novo método do gradiente conjugado espectral (*spectral conjugate gradient*, SCG) (Wang *et al.*, 2020). O novo esquema de escolha dos parâmetros espectral e conjugado favorece o desenvolvimento de uma nova direção de busca que satisfaz a propriedade espectral e a condição de descida simultaneamente, favorecendo alto ganho funcional a cada iteração. Além disso, neste novo algoritmo, emprega-se o uso de informação de segunda ordem por meio da aproximação de Broyden-Fletcher-Goldfarb-Shanno (BFGS) (Babaie-Kafaki, 2015) para a Hessiana da função objetivo dual o que contribui para uma rápida convergência sem grandes custos de processamento por iteração.

Com o intuito de esparsificar as soluções obtidas por essa proposta, foi desenvolvido uma metodologia de poda iterativa baseada na proximidade de um padrão ao hiperplano de decisão. Conforme as atualizações nos valores dos multiplicadores de Lagrange são realizadas, avalia-se a proximidade de cada padrão ao hiperplano, selecionando aqueles com maior proximidade e adicionado-os ao conjunto ativo de multiplicadores que passarão por atualizações subsequentes, os demais não serão atualizados.

Dessa maneira, o conjunto ativo cresce a cada iteração e a métrica de validação tende a crescer até que ocorra a estabilização do algoritmo, sendo que nesse caso o processo é interrompido. Portanto, obtém-se uma metodologia para poda com tamanho variável já que não é necessário definir um número final de vetores-suporte, o próprio algoritmo estabiliza em uma quantidade ótima, contribuindo assim para a redução no número de hiperparâmetros no estágio de ajuste do modelo.

1.1 Estudo Bibliométrico

Nesta seção, apresenta-se um estudo bibliométrico sobre a temática dos modelos SVM, LSSVM e LSSVM esparsa considerando tanto artigos de revista como de conferência nos últimos cinco anos desconsiderando o ano de 2025, ou seja, de 2019 até 2024. Destaca-se que a base de dados utilizada foi a da SCOPUS (Elsevier). A ideia é avaliar o impacto do tema nos últimos anos, considerando número de publicações, principais pesquisadores e avaliação de tendências nos temas de pesquisa. a Análise é feita via o *framework bibliometrix* (Derviş, 2019) desenvolvido para a linguagem R (Team, 2000). Um resumo da busca é dado na Tabela 1.

Tabela 1 – Detalhamento sobre a consulta realizada à base da SCOPUS.

Tópicos	**Informações**
Escopo	Impacto dos Temas 'SVM', 'LSSVM' sobre a área de aprendizado de máquina.
Base de Dados	SCOPUS
Termos Pesquisados	"support vector machine"or "least squares support vector machine"or "sparse least squares support vector machine".
Restrições	– Intervalo: Últimos 5 anos (desconsiderando o ano de 2025); – Documentos: artigos científicos e de revisão; – Áreas pesquisadas: aprendizado de máquina e inteligência artificial.
Ferramenta de Análise	*software bibliometrix*

Fonte: Elaborada pelo autor.

Ao realizar a pesquisa, o número total de documentos obtidos foi de 103.199 documentos, no entanto, para a exportação da base de dados, a plataforma da SCOPUS limita o número de documentos a ser exportado para um valor de 20.000 documentos, com isso as análises presentes nesta tese são limitadas aos 20.000 documentos mais recentes obtidos na pesquisa, o que corresponde aos anos de 2023 e 2024. Os números de documentos publicados em cada ano é reportada na Tabela 2.

Tabela 2 – Número de documentos publicados por ano.

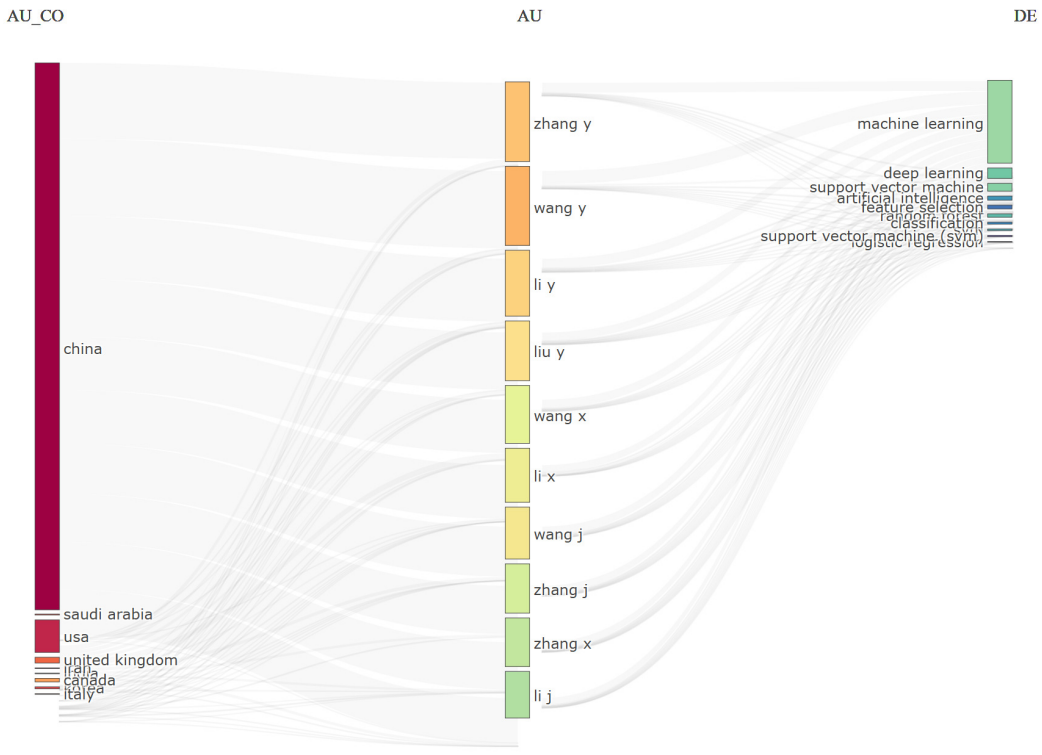
Ano	**Quantidade**
2024	18835
2023	1165

Fonte: Elaborada pelo autor.

Estes achados indicam uma taxa de crescimento anual de 1.516,74%, além disso, os documentos recuperados trouxeram 161.772 referências, 37.134 palavras-chave e envolveram 57.807 autores com uma proporção de 5 autores por documento. Do total de coautores cerca de 22,34% são de países distintos do autor principal, indicando uma baixa internacionalização na colaboração de pesquisas dentro destes temas de busca.

A Figura 1 apresenta os países de origem dos 10 autores mais citados dentro da busca, o nome de citação de cada autor e as 10 palavras chaves que ocorrem com mais frequência na base recuperada. Palavras chave '*machine learning*', '*deep learning*' e '*support vector machine*' foram as mais frequentes dentro da base de documentos recuperada, ilustrando a importância e o impacto que estes temas ainda apresentam dentro da área de inteligência artificial.

Figura 1 – Fluxograma de informações sobre os 10 principais autores na busca.

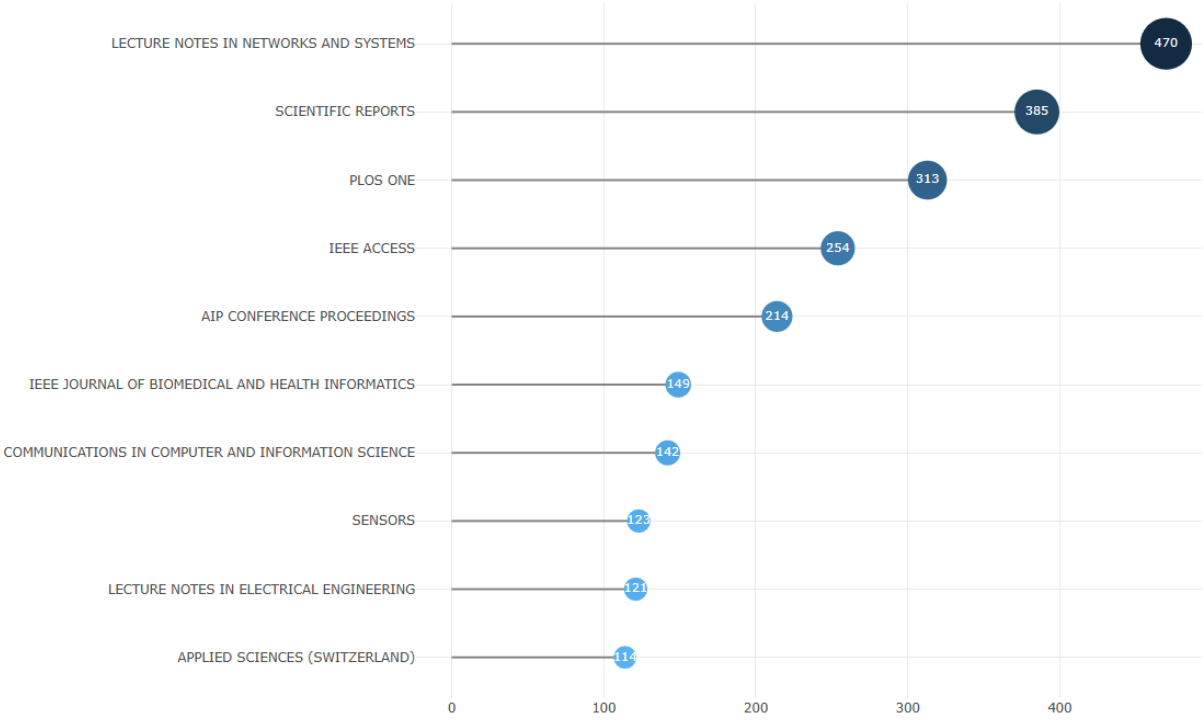


Fonte: Elaborada pelo autor.

Relativo ao número de documentos publicados, a Figura 2 traz os periódicos com maior número de documentos publicados.

Por fim, a Figura 3 ilustra um gráfico de árvore com as principais palavras chave dos documentos recuperados da base. Novamente, tem-se que por mais que seja um modelo

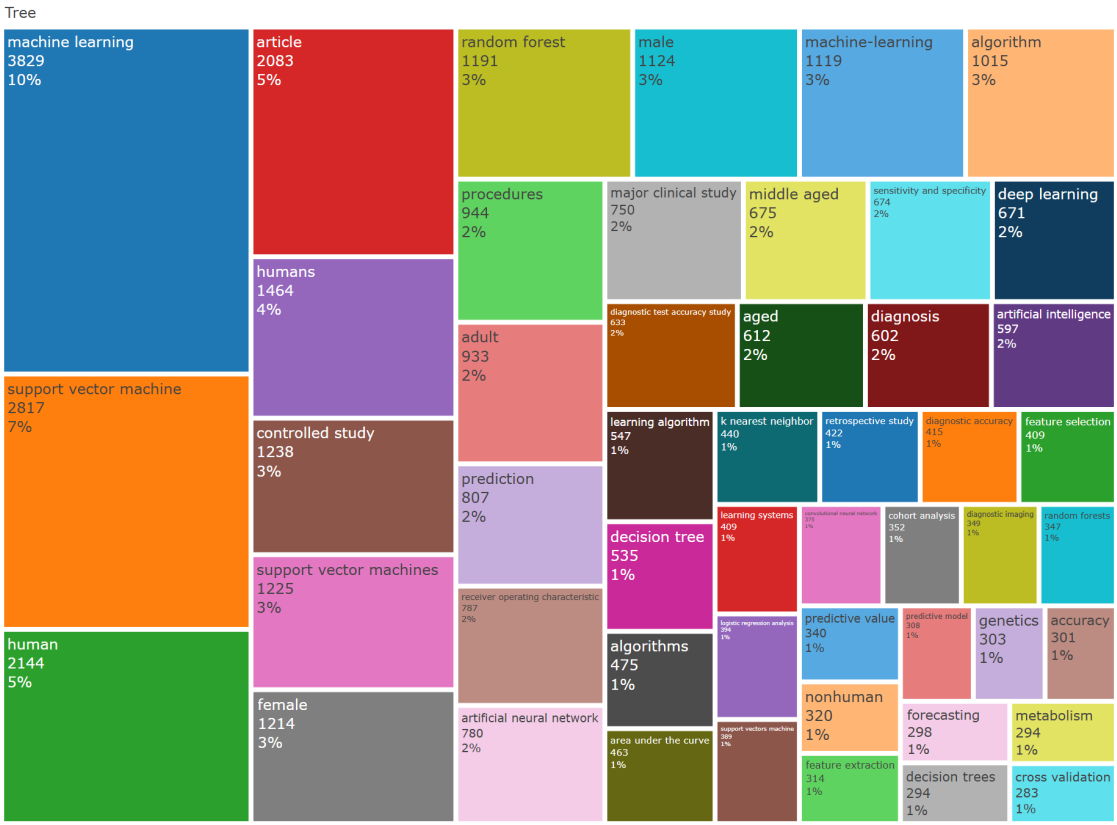
Figura 2 – Periódicos em termos do número de documentos publicados.



Fonte: Elaborada pelo autor.

que surgiu no final dos anos 90, o SVM ainda continua sendo um tópico significativo na área de aprendizado de máquina, tanto do ponto de vista teórico como aplicado, ratificando assim a importância do tema estudado na tese.

Figura 3 – Árvore hierárquica ilustrando as palavras chave mais frequentes do documentos recuperados na consulta.



Fonte: Elaborada pelo autor.

1.2 Objetivo Geral e Específicos

O objetivo principal desta tese é a proposição de novas metodologias para o treinamento rápido e esparso de modelos LSSVM, bem como, a validação de tais propostas por meio de extensivas simulações numéricas em bases de dados reais e artificiais, considerando ambos os cenários de alta e baixa volumetria de dados.

Quanto aos objetivos específicos desta tese, pode-se destacar os seguintes itens:

- Realizar uma revisão bibliográfica sobre o tema em análise;
- Apresentar e avaliar as metodologias clássicas e bem estabelecidas para o treinamento rápido e esparso de modelos LSSVM;
- Apresentar a heurística e a fundamentação teórica para cada proposta considerando ambas análise de convergência e de complexidade dos algoritmos resultantes;
- Avaliar o desempenho das novas propostas tanto no que diz respeito a capacidade preditiva, nível de esparsidade promovido, processamento de grandes bases de dados e qualidade da

fronteira de decisão gerada;

– Avaliar o desempenho das novas propostas em uma grande base de dados real, envolvendo o problema de previsão de irradiância solar de curto prazo.

1.3 Produção Científica

Ao longo da construção desta tese de doutorado, os seguintes artigos foram desenvolvidos:

1. **Marinho, F. P.**, Almeida, W. D., Santos, V. O., Rocha Neto, A. R. (2025). A Fast Least Square SVM Training Approach via Spectral Conjugate Gradient. **Pattern Analysis and Applications** (submetido).
2. **Marinho, F. P.**, Almeida, W. D., Santos, V. O., Rocha Neto, A. R. (2025). A Fast Sparse LSSVM Training Algorithm: A Dual Approach. **Applied Intelligence** (submetido).
3. Sousa, F. A. **Marinho, F. P.**, Rocha, T. A., Rocha Neto, A. R. (2025). New Distance-based Training Algorithms for Support Vector Machines. **Logic Journal of IGPL** (submetido por convite).
4. **Marinho, F. P.**, Almeida, W. D., Santos, V. O., Rocha Neto, A. R. (2025). A New Approach for Obtain Reduced Sets in Least Squares Support Vector Machine: Lenghten via Levenberg-Marquardt. **35th Brazilian Conference on Intelligent Systems (BRACIS)** (aprovado).
   Disponível em: https://doi.org/10.1007/978-3-032-15987-8_6
5. **Marinho, F. P.**, Almeida, W. D., Santos, V. O., Rocha Neto, A. R. (2024). Sparse Least Square SVM in Primal via Nesterov Accelerated Alternating Directions Method of Multipliers. **18th International Work-Conference on Artificial Neural Networks (IWANN)** (aprovado).
   Disponível em: https://doi.org/10.1007/978-3-032-02725-2_7
6. **Marinho, F. P.**, Rocha P. A. C., Rocha Neto, A. R., Bezerra, F. D. V. (2022). Short-Term Solar Irradiance Forecasting Using CNN-1D, LSTM and CNN-LSTM Deep Neural Networks: A Case Study with the Folsom (USA) Dataset. **Journal of Solar Energy Engineering**, v. 145, n. 4, p. 041002.
   Disponível em: https://doi.org/10.1115/1.4056122
7. **Marinho, F. P.**, Santos, V. O., Rocha P. A. C., Rocha Neto, A. R. (2024). Forecasting Global and Direct Solar Irradiance with Machine Learning Algorithms: Insights from

Recursive Feature Selection and SHAP Analysis. **20th Brazilian Congress of Thermal Sciences and Engineering**.

1.4 Produção Científica não Abordada na Tese

Além dos artigos citados anteriormente, também foram desenvolvidos outros trabalhos durante o doutorado, que embora abordem temas e aplicações relacionadas, não são tratados nesta tese por questão de limitação do tamanho do documento.

1. Bezerra, G. C. B., **Marinho F. P.**, Rocha Neto, A. R. (2025). A New Machine Learning Approach to Detect Student Success in Pair Programming. **XVII Congresso Brasileiro de Inteligência Computacional**
2. Santos, V. O., **Marinho, F. P.**, Rocha P. A. C., Thé, J. V. G., Gharabaghi, B. (2024). Application of Quantum Neural Network for Solar Irradiance Forecasting: A Case Study Using the Folsom Dataset, California. **Energies**, v. 17, n. 14, p. 3580.
   Disponível em: https://doi.org/10.3390/en17143580
3. **Marinho, F. P.**, Rocha P. A. C., Rocha Neto, A. R., Santos, V. O. (2024). Dimensional reduction for solar irradiance forecasting problem using principal components analysis and Turk-Pentland strategy. **International Joint Conference on Neural Networks (IJCNN)**.
   Disponível em: 10.1109/IJCNN60899.2024.10651398
4. Bezerra, F. D. V., **Marinho, F. P.**, Rocha P. A. C., Santos, V. O., Thé, J. V. G., Gharabaghi, B. (2023). Machine Learning Dynamic Ensemble Methods for Solar Irradiance and Wind Speed Predictions. **Atmosphere**, v. 14, n. 11, p. 1635.
   Disponível em: https://doi.org/10.3390/atmos14111635
5. **Marinho, F. P.**, Rocha P. A. C., Rocha Neto, A. R. (2022). Previsão de Irradiância Solar de Curto Prazo Utilizando Modelo de Envelopes para os Preditores. **XXIV Congresso Brasileiro de Automática**.
   Disponível em: https://sba.org.br/open_journal_systems/index.php/cba/article/view/3473

1.5 Organização da Tese

O restante da tese está organizada nos seguintes capítulos.

O **Capítulo 2** apresenta a fundamentação teórica sobre modelos LSSVM, apresentando a formulação de seu problema primal e dual, bem como, as principais metodologias para a

sua solução incluindo o método da inversa, pseudo-inversa, solução via gradiente conjugado de Hesteness-Stiefel e solução via algoritmo de Levenberg-Marquardt.

O **Capítulo 3** apresenta as principais variantes esparsas do modelo LSSVM, além de, apresentar as metodologias empregadas para a solução do problema dual da LSSVM, no caso, o SMO de primeira e segunda ordem e o método CSMO.

No **Capítulo 4** as duas propostas são discutidas em termos de formulação matemática, análise de convergência e complexidade. Também se discute as principais contribuições proporcionadas pelas mesmas, além de abordar os principais pontos de melhorias a serem implementados em estudos futuros.

O **Capítulo 5** discute a metodologia para a realização das simulações numéricas, apresentando as bases de dados consideradas, a estratégia para o ajuste dos modelos e o *pipeline* de dados utilizado na obtenção dos resultados. Os resultados obtidos para cada proposta, são avaliados e discutidos, onde se considerou quatro aspectos principais: Capacidade preditiva de cada modelo, nível de esparsidade promovido, qualidade da fronteira de decisão gerada e capacidade de processar bases de grande escala. Com isso, as discussões dos resultados para cada proposta são realizadas em seções específicas com o intuito de facilitar a leitura e avaliação por parte dos leitores desta tese.

Por fim, o **Capítulo 6** apresenta um resumo dos resultados obtidos, além de, trazer as principais conclusões e contribuições da tese, resumindo os principais desafios e melhorias que podem ser realizadas em cada proposta em estudos futuros.

2 FUNDAMENTOS TEÓRICOS DO MODELO LSSVM

As máquinas de vetores-suporte de mínimos quadrados representam uma técnica de aprendizado supervisionado derivada das clássicas máquinas de vetores-suporte, na qual é capaz de solucionar problemas de classificação de padrões, por meio dos LSSVCs, e de aproximação de funções utilizando os LSSVRs (Suykens e Vandewalle, 1999).

A LSSVM difere da SVM em sua abordagem de otimização, a qual consiste em resolver o problema primal considerando apenas restrições de igualdade com a adição de termos de erros, que atuam como variáveis de folga tolerando erros de classificação e desvios relativos à variável alvo numérica. As mudanças na formulação do problema primal tornam o processo de solução mais simples, uma vez que, o treinamento da LSSVM é realizado com base na resolução de um sistema linear KKT.

2.1 Formulação Primal

Dado o conjunto de treinamento $\mathscr{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ de N amostras, em que $\boldsymbol{x}_i \in \mathbb{R}^p$ e $y_i \in \{-1, 1\}$. Sejam também $\mathscr{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ o conjunto de amostras de entrada e $\mathscr{Y} = \{y_i\}_{i=1}^{N}$ o conjunto correspondente de rótulos. O problema primal do LSSVC é dado

$$\begin{aligned} min_{\boldsymbol{w},\boldsymbol{\varepsilon}} \quad & \tau(\boldsymbol{w},\boldsymbol{\varepsilon}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N}\varepsilon_i^2, \\ \text{s.t.} \quad & y_i[\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) + b] = 1 - \varepsilon_i, \;\; i = 1, \ldots, N. \end{aligned} \tag{2.1}$$

A função $\boldsymbol{\phi} : \mathbb{R}^p \longrightarrow \mathbb{R}^S$ é um mapeamento para um espaço de atributos de alta dimensionalidade [1], $\boldsymbol{\varepsilon} = \{\varepsilon_i\}_{i=1}^{N}$ são os erros e $\gamma \in \mathbb{R}^+$ é um parâmetro de custo que controla o equilíbrio entre permitir erros de treinamento e forçar margens rígidas. A função Lagrangeana para o problema primal do LSSVC é dada por

$$\mathscr{L}(\boldsymbol{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}) = \tau(\boldsymbol{w},\boldsymbol{\varepsilon}) + \sum_{i=1}^{N}\alpha_i(1 - \varepsilon_i - y_i[\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) + b]), \tag{2.2}$$

em que $\boldsymbol{\alpha} = \{\alpha\}_{i=1}^{N}$ são multiplicadores de Lagrange com $\alpha_i \in \mathbb{R}$. As condições de otimalidade de KKT são apresentadas como se segue

$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{w}} = 0 \longrightarrow \boldsymbol{w} = \sum_{i=1}^{N}\alpha_i y_i \boldsymbol{\phi}(\boldsymbol{x}_i), \tag{2.3}$$

[1] Podendo ser infinitamente dimensional, ou seja, $S = \infty$

$$\frac{\partial \mathscr{L}}{\partial b} = 0 \longrightarrow \sum_{i=1}^{N} \alpha_i y_i = 0, \tag{2.4}$$

$$\frac{\partial \mathscr{L}}{\partial \varepsilon_i} = 0 \longrightarrow \alpha_i = \gamma \varepsilon_i \quad i = 1, \ldots, N, \tag{2.5}$$

$$\frac{\partial \mathscr{L}}{\partial \alpha_i} = 0 \longrightarrow y_i[\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b] = 1 - \varepsilon_i \quad i = 1, \ldots, N. \tag{2.6}$$

Substituindo os valores de $\boldsymbol{w}$, Equação (2.3), e $\boldsymbol{\varepsilon}$, Equação (2.5), na condição referente a Equação (2.6), obtém-se a expressão dada por

$$y_i[\sum_{j=1}^{N} \alpha_j y_j \phi^T(\boldsymbol{x}_j)\phi(\boldsymbol{x}_i) + b] = 1 - \frac{\alpha_i}{\gamma} \quad i = 1, \ldots, N, \tag{2.7}$$

além disso, definindo $K_{ji} = \phi^T(\boldsymbol{x}_j)\phi(\boldsymbol{x}_i)$, tem-se que a expressão anterior pode ser simplificada, resultando na formulação indicada a seguir

$$y_i[\sum_{j=1}^{N} \alpha_j y_j \boldsymbol{K}_{ji} + b] = 1 - \frac{\alpha_i}{\gamma} \longrightarrow \sum_{j=1}^{N} \alpha_j y_i y_j \boldsymbol{K}_{ji} + y_i b = 1 - \frac{\alpha_i}{\gamma} \quad i = 1, \ldots, N. \tag{2.8}$$

Considerando $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$, $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ e $\boldsymbol{Y} = diag(y_1, \ldots, y_N)$, tem-se que a Equação (2.8) pode ainda ser dada em formato matricial, resultando em

$$\boldsymbol{YKY\alpha} + b\boldsymbol{y} = \boldsymbol{1} - \boldsymbol{I}\frac{\boldsymbol{\alpha}}{\gamma}, \tag{2.9}$$

ou ainda

$$\left(\boldsymbol{YKY} + \frac{\boldsymbol{I}}{\gamma}\right)\boldsymbol{\alpha} + b\boldsymbol{y} = \boldsymbol{1}. \tag{2.10}$$

Juntando com a condição adicional $\boldsymbol{y}^T\boldsymbol{\alpha} = 0$, Equação (2.4), as duas expressões podem ser condensadas no sistema linear $\boldsymbol{Au} = \boldsymbol{v}$, dado por

$$\left[\begin{array}{c|c} 0 & \boldsymbol{y}^T \\ \hline \boldsymbol{y} & \boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I} \end{array}\right] \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{1} \end{bmatrix}, \tag{2.11}$$

em que $\boldsymbol{\Omega}$ é a matriz dada por $\Omega_{ij} = y_i y_j \boldsymbol{K}_{ij}$.

Para a derivação do LSSVR, o racional é similar ao caso para classificação, onde as saídas serão consideradas valores numéricos ao invés de apenas os rótulos binários $\{+1,-1\}$. Sendo $y = \boldsymbol{w}^T\phi(\boldsymbol{x})+b$, com $\boldsymbol{x}\in\mathbb{R}^n$ e $y\in\mathbb{R}$, $\phi(.)$ sendo o mapeamento como definido no caso de classificação e dado o conjunto de treinamento $\mathscr{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ de N amostras, então o problema de otimização primal é dado por

$$\begin{aligned} min_{\boldsymbol{w},\boldsymbol{\varepsilon}} \quad & \tau(\boldsymbol{w},\boldsymbol{\varepsilon}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N}\varepsilon_i^2, \\ \text{s.t.} \quad & \boldsymbol{w}\phi(\boldsymbol{x}_i)+b = y_i-\varepsilon_i, \;\; i=1,\ldots,N. \end{aligned} \tag{2.12}$$

A função Lagrangeana para o problema primal do LSSVR é como se segue

$$\mathscr{L}(\boldsymbol{w},b,\boldsymbol{\varepsilon},\boldsymbol{\alpha}) = \tau(\boldsymbol{w},\boldsymbol{\varepsilon}) + \sum_{i=1}^{N}\alpha_i(y_i-\boldsymbol{w}^T\phi(\boldsymbol{x}_i)-b-\varepsilon_i). \tag{2.13}$$

Neste cenário, após a eliminação de $\boldsymbol{w}$ e $\boldsymbol{\varepsilon}$ e pelo uso das condições de otimalidade de KKT de maneira análoga a derivação para o LSSVC, chega-se no sistema linear $\boldsymbol{Au} = \boldsymbol{v}$

$$\left[\begin{array}{c|c} 0 & \mathbf{1}^T \\ \hline \mathbf{1} & \boldsymbol{K}+\gamma^{-1}\boldsymbol{I} \end{array}\right]\left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \boldsymbol{y} \end{array}\right]. \tag{2.14}$$

Em ambos os casos, a matriz $\boldsymbol{K}$ é a matriz de *kernel*, onde $K_{i,j} = \langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle = \mathscr{K}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ para $i,j = 1,2,\ldots,N$ com $\mathscr{K}(.,.)$ sendo uma função de *kernel* de Mercer, γ é o parâmetro de custo, $\boldsymbol{I}$ é a matriz identidade de dimensão $N\times N$, e $\mathbf{1}$ é a matriz de uns com dimensão adequada. Das condições KKT, obtem-se

$$\boldsymbol{w}_{LSSVC} = \sum_{i=1}^{N}\alpha_i y_i\phi(\boldsymbol{x}_i), \tag{2.15}$$

$$\boldsymbol{w}_{LSSVR} = \sum_{i=1}^{N}\alpha_i\phi(\boldsymbol{x}_i), \tag{2.16}$$

$$\alpha_i = y_i\varepsilon_i. \tag{2.17}$$

A partir da Equação (2.17), surge o principal problema relacionado à LSSVM: **a falta de esparsidade**. Em problemas do mundo real, ε_i geralmente é não nulo e $y_i\in\{+1,-1\}$, resultando em $\alpha_i\neq 0$. O efeito prático dessa construção é que a maioria das amostras de treinamento serão utilizadas como vetores-suporte.

2.2 Formulação Dual

A derivação do problema dual é feita considerando o problema dado pela Equação (2.12). Neste ponto, vale frisar que foi utilizado o problema primal para treinamento do LSSVR, no entanto um procedimento análogo pode ser aplicado para o caso do LSSVC, onde o problema dual resultante é equivalente ao obtido pela derivação aqui realizada. Ao utilizar as condições KKT, obtem-se as seguintes seguintes expressões

$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{w}} = 0 \longrightarrow \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \phi(\boldsymbol{x}_i), \tag{2.18}$$

$$\frac{\partial \mathscr{L}}{\partial b} = 0 \longrightarrow \sum_{i=1}^{N} \alpha_i = 0, \tag{2.19}$$

$$\frac{\partial \mathscr{L}}{\partial \varepsilon_i} = 0 \longrightarrow \alpha_i = \gamma \varepsilon_i \quad i = 1, \ldots, N, \tag{2.20}$$

$$\frac{\partial \mathscr{L}}{\partial \alpha_i} = 0 \longrightarrow \boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b = y_i - \varepsilon_i \quad i = 1, \ldots, N. \tag{2.21}$$

Substituindo as condições dadas pelas Equações (2.18) e (2.20) na Lagrangeana do problema, como mostrada na Equação (2.13) e com a realização de algumas manipulações com somatórios é possível obter a expressão resultante, dada por

$$\begin{aligned}
\mathscr{L}(\boldsymbol{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N}\varepsilon_i^2 + \sum_{i=1}^{N}\alpha_i(y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b - \varepsilon_i) \\
&= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N}\varepsilon_i^2 - \boldsymbol{w}^T\sum_{i=1}^{N}\alpha_i\phi(\boldsymbol{x}_i) - b\sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\varepsilon_i + \sum_{i=1}^{N}\alpha_i y_i \\
&= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}(\frac{1}{\gamma^2})\sum_{i=1}^{N}\alpha_i^2 - \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i^2 + \sum_{i=1}^{N}\alpha_i y_i \\
&= -\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \frac{1}{2\gamma}\sum_{i=1}^{N}\alpha_i^2 + \sum_{i=1}^{N}\alpha_i y_i.
\end{aligned} \tag{2.22}$$

Utilizando as expressões $\boldsymbol{w} = \sum_{i=1}^{N}\alpha_i\phi(\boldsymbol{x}_i)$ e $\sum_{i=1}^{N}\alpha_i = 0$ é possível obter a Lagrangeana em função apenas da variável dual, como indicado em

$$\mathscr{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\phi^T(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j) - \frac{1}{2\gamma}\sum_{i=1}^{N}\alpha_i^2 + \sum_{i=1}^{N}\alpha_i y_i. \tag{2.23}$$

Relembrando que os elementos da matriz de kernel $\boldsymbol{K}$ são dados por $\boldsymbol{K}_{ij} = \phi^T(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)$, então problema dual resultante é dado por

$$\begin{aligned} max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K_{ij} - \frac{1}{2\gamma}\sum_{i=1}^{N}\alpha_i^2 + \sum_{i=1}^{N}\alpha_i y_i \\ s.t. \quad & \sum_{i=1}^{N}\alpha_i = 0, \end{aligned} \tag{2.24}$$

além disso, considerando os vetores $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$ e lembrando que maximizar uma função objetivo é equivalente a minimizar seu negativo, então é possível escrever a expressão anterior em formato matricial, como dado a seguir

$$\begin{aligned} min_{\boldsymbol{\alpha}} \quad & \mathscr{D}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} + \frac{1}{2\gamma}\boldsymbol{\alpha}^T\boldsymbol{I}\boldsymbol{\alpha} - \boldsymbol{y}^T\boldsymbol{\alpha} \\ s.t. \quad & \sum_{i=1}^{N}\alpha_i = 0, \end{aligned} \tag{2.25}$$

ou ainda, como

$$\begin{aligned} min_{\boldsymbol{\alpha}} \quad & \mathscr{D}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y}^T\boldsymbol{\alpha} \\ s.t. \quad & \sum_{i=1}^{N}\alpha_i = 0, \end{aligned} \tag{2.26}$$

em que $\tilde{\boldsymbol{K}} = \boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}$ e $\mathscr{D}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y}^T\boldsymbol{\alpha}$ representa a função objetivo para o problema dual.

2.3 Solução Baseada na Matriz Inversa

A solução para a LSSVM pode ser obtida pela simples inversão da matriz $\boldsymbol{A}$ no sistema KKT correspondente, desde que a mesma seja não singular, como indicado a seguir

$$\begin{aligned} \boldsymbol{A}\boldsymbol{u} &= \boldsymbol{v} \\ \boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{u} &= \boldsymbol{A}^{-1}\boldsymbol{v} \\ \boldsymbol{u} &= \boldsymbol{A}^{-1}\boldsymbol{v}. \end{aligned} \tag{2.27}$$

O problema dessa abordagem está na ausência de garantias de invertibilidade da matriz $\boldsymbol{A}$, além disso, o cálculo de inversas de matrizes podem apresentar diversas dificuldades numéricas, como crescimento exponencial na complexidade computacional com o aumento das dimensões da matriz de entrada e instabilidades numéricas (Golub e Loan, 2013). Portanto, por mais que seja o procedimento mais simples, não é o mais comumente utilizado e também não foi o empregado nesta tese.

2.4 Solução Baseada na Pseudo Inversa

Para os casos em que a matriz $\boldsymbol{A}$ é não-quadrada, sabe-se que a mesma é singular e uma alternativa é o emprego de inversas generalizadas, como a matriz pseudo-inversa de Moore-Penrose (Golub e Loan, 2013; Penrose, 1955; Ben-Israel e Greville, 2006).

Neste caso, a solução obtida equivale a considerar cada uma das colunas excluídas como possuindo multiplicador de Lagrange associado com valor zero. A remoção de linhas, porém, equivale à remoção das restrições associadas ao problema de otimização (Leao e Neto, 2022). Assim, a obtenção do vetor solução $\boldsymbol{u}$ para uma matriz não-quadrada $\boldsymbol{A}$ é dada por

$$\boldsymbol{u} = \boldsymbol{A}^{\dagger}\boldsymbol{v}, \tag{2.28}$$

em que

$$\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T. \tag{2.29}$$

2.5 Solução Baseada no Gradiente Conjugado de Hesteness-Stiefel

2.5.1 Métodos de Descida

Os métodos de descida obtêm soluções aproximadas de sistemas lineares reformulando-os como um problema de minimização de uma forma quadrática. De fato, considerando a função $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ dada por

$$J(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T\boldsymbol{A}\boldsymbol{u} - \boldsymbol{u}^T\boldsymbol{v}. \tag{2.30}$$

Supondo ainda que a matriz $\boldsymbol{A}$ é simétrica e definida positiva, tem-se que a função $\boldsymbol{J}$ é estritamente convexa e admite um mínimo absoluto no ponto que torna seu gradiente nulo, ou seja, no vetor que satisfaz

$$\nabla_{\boldsymbol{u}}\boldsymbol{J}(\boldsymbol{u}) = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{u} - \boldsymbol{v} = \boldsymbol{A}\boldsymbol{u} - \boldsymbol{v} = 0, \tag{2.31}$$

ou ainda

$$\nabla_{\boldsymbol{u}}\boldsymbol{J}(\boldsymbol{u}) = 0 \longrightarrow \boldsymbol{A}\boldsymbol{u} = \boldsymbol{v}. \tag{2.32}$$

Portanto, o vetor que minimiza $\boldsymbol{J}(.)$ é exatamente a solução de $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{v}$.

Para a solução do sistema linear, os métodos de descida iniciam de uma aproximação inicial da solução $\boldsymbol{u}_0$ e geram uma sequência de iteradas $\{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ satisfazendo a condição

de $\boldsymbol{J}(\boldsymbol{u}_{k+1}) \leq \boldsymbol{J}(\boldsymbol{u}_k)$, caracterizando-os como métodos iterativos. A obtenção de $\boldsymbol{u}_{k+1}$ a partir de $\boldsymbol{u}_k$ envolve dois fatores principais (Watkins, 2004):

- Seleção/construção de uma direção de descida;
- Realização de uma busca em linha na direção de descida correspondente.

A seleção da direção de busca corresponde a determinar o vetor $\boldsymbol{z_k}$ que indica a direção de variação sobre o vetor $\boldsymbol{u}_k$ para a obtenção de $\boldsymbol{u}_{k+1}$. Existem várias estratégias para a construção da direção de descida, em que cada uma define um tipo particular de método de descida, por exemplo quando $z_k = \boldsymbol{r}_k = \boldsymbol{v} - \boldsymbol{A}\boldsymbol{u}_k$, ou seja, a direção de descida é o próprio resíduo do sistema em uma dada iteração, tem-se que o método de descida é o gradiente descendente (Watkins, 2004).

Uma vez que se tenha a direção de descida, $\boldsymbol{u}_{k+1}$ é selecionado como um ponto sobre a linha de busca $\{\boldsymbol{u}_k + \alpha_k \boldsymbol{z}_k | \, \alpha_k \in \mathbb{R}\}$, neste caso é válida a relação dada por

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \alpha \boldsymbol{z}_k. \tag{2.33}$$

Obviamente, existem infinitos pontos sobre a linha, mas se deseja aquele que fornece um menor valor para a função objetivo $\boldsymbol{J}(.)$, de tal forma que vale a desigualdade $\boldsymbol{J}(\boldsymbol{u}_{k+1}) \leq \boldsymbol{J}(\boldsymbol{u}_k)$. Uma maneira de conseguir satisfazer tal critério seria pela escolha de um passo de aprendizado α, tal que $\alpha = \min_{\alpha \in \mathbb{R}} \boldsymbol{J}(\boldsymbol{u}_k + \alpha z_k)$. Esta metodologia para seleção de α é ainda denominada de busca em linha exata (Watkins, 2004; Wright *et al.*, 1999).

Para algumas funções $\boldsymbol{J}(.)$ o trabalho envolvido no cálculo de α pode ser considerável. Felizmente, para a forma quadrática dada pela Equação (2.30) o processo é relativamente simples e dado pela expressão

$$\phi'(\alpha) = \frac{\mathrm{d}\boldsymbol{J}(\boldsymbol{u}_k + \alpha \boldsymbol{z}_k)}{\mathrm{d}\alpha} = \nabla \boldsymbol{J}^T(\boldsymbol{u}_k + \alpha \boldsymbol{z}_k) \boldsymbol{z}_k = 0, \tag{2.34}$$

ou ainda

$$[\boldsymbol{A}(\boldsymbol{u}_k + \alpha \boldsymbol{z}_k) - \boldsymbol{v}]^T \boldsymbol{z}_k = 0 \longrightarrow -\boldsymbol{r}_k^T \boldsymbol{z}_k + \alpha \boldsymbol{z}_k^T \boldsymbol{A} \boldsymbol{z}_k = 0, \tag{2.35}$$

resultando em

$$\alpha = \frac{\boldsymbol{r}_k^T \boldsymbol{z}_k}{\boldsymbol{z}_k^T \boldsymbol{A} \boldsymbol{z}_k}, \tag{2.36}$$

em que $\boldsymbol{r}_k = \boldsymbol{v} - \boldsymbol{A}\boldsymbol{u}_k$ é o resíduo do sistema na k-ésima iteração.

2.5.2 *Método dos Gradientes Conjugados*

O método dos gradientes conjugados é um exemplo de método de descida e assim como no caso do gradiente descendente, suas direções de busca são construídas a partir dos resíduos do sistema linear ao longo das iterações. A diferença está no fato, de que ao invés de utilizar os próprios resíduos como direções de descida, o método dos gradientes conjugados utilizam um conjunto de direções $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_n$ construídas por um processo de conjugação de Gram-Schmidt dos resíduos (Shewchuk, 1994).

Neste caso, considerando o conjunto de vetores $\{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n-1}\}$ é possível obter um novo conjunto de vetores $\{\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n-1}\}$ que são dois a dois $\boldsymbol{A}$-ortogonais, isto é

$$\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_j = 0, \ \forall i, j \in \{0, \ldots, n-1\}. \tag{2.37}$$

O procedimento é iniciado fazendo $\boldsymbol{d}_0 = \boldsymbol{u}_0$ onde os demais vetores são obtidos pela recorrência que se segue

$$\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_j = \boldsymbol{u}_i^T \boldsymbol{A} \boldsymbol{d}_j + \sum_{k=0}^{i-1} \beta_{ik} \boldsymbol{d}_k^T \boldsymbol{A} \boldsymbol{d}_j, \tag{2.38}$$

em que,

$$\beta_{ik} = -\frac{\boldsymbol{u}_i^T \boldsymbol{A} \boldsymbol{d}_j}{\boldsymbol{d}_j^T \boldsymbol{A} \boldsymbol{d}_j} \tag{2.39}$$

Esta escolha traz alguns benefícios interessantes. Pelo fato dos resíduos serem ortogonais às direções de busca de iterações passadas, isto implica que as novas direções são ainda linearmente independentes às direções passadas. Além disso, como os vetores de busca são construídos pela conjugação dos resíduos e estes são ortogonais às direções de busca de iterações anteriores, estes também são normais aos resíduos de iterações passadas, ou seja

$$\boldsymbol{r}_i^T \boldsymbol{r}_j = 0, \ \forall i, j \in \{0, \ldots, n-1\} \tag{2.40}$$

Seja $\mathscr{A}_i$ o subespaço vetorial de dimensão i, tal que $\mathscr{A}_i = span\{\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{i-1}\}$[2], tem-se que para o caso dos gradientes conjugados este ainda pode ser dado como $\mathscr{A}_i = span\{\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{i-1}\}$ e sendo

$$\boldsymbol{r}_{i+1} = \boldsymbol{A}\boldsymbol{u}_{i+1} - \boldsymbol{A}\boldsymbol{u} = -\boldsymbol{A}\boldsymbol{e}_{i+1}, \tag{2.41}$$

[2] Dizer que $\mathscr{A}_i = span\{\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{i-1}\}$, significa que o subespaço $\mathscr{A}_i$ é gerado pelo conjunto de vetores $\{\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{i-1}\}$.

em que $\boldsymbol{e}_{i+1} = \boldsymbol{u} - \boldsymbol{u}_{i+1}$ representa o erro da solução aproximada na $(i+1)$-ésima iteração. Portanto, tem-se que

$$\boldsymbol{r}_{i+1} = -\boldsymbol{A}\boldsymbol{e}_{i+1} = -\boldsymbol{A}(\boldsymbol{e}_i + \alpha_i \boldsymbol{d}_i), \tag{2.42}$$

resultando em

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{A}\boldsymbol{d}_i. \tag{2.43}$$

Do exposto é possível ainda inferir que o resíduo $\boldsymbol{r}_i$ é uma combinação linear de resíduos anteriores e de $\boldsymbol{A}\boldsymbol{d}_{i-1}$. Portanto os subespaços $\mathscr{A}_i$ para o caso dos gradientes conjugados ainda satisfazem a seguinte definição

$$\mathscr{A}_i = span\{\boldsymbol{d}_0, \boldsymbol{A}\boldsymbol{d}_0, \boldsymbol{A}^2\boldsymbol{d}_0 \ldots, \boldsymbol{A}^{i-1}\boldsymbol{d}_0\}, \tag{2.44}$$

ou ainda,

$$\mathscr{A}_i = span\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0 \ldots, \boldsymbol{A}^{i-1}\boldsymbol{r}_0\}. \tag{2.45}$$

Os subespaços gerados pela aplicação sucessiva de uma matriz $\boldsymbol{A}$, são chamados de subespaços de Krylov e o método dos gradientes conjugados é ainda de uma classe mais geral de métodos iterativos de Krylov (Watkins, 2004; Trefethen e Bau, 2022; Faghih *et al.*, 2025).

Da definição do termo $\beta_{ik} = -\frac{\boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j}{\boldsymbol{d}_j^T \boldsymbol{A}\boldsymbol{d}_j}$ no processo de conjugação de Gram-Schmidt e da Equação (2.43), fazendo o produto interno desta por $\boldsymbol{r}_i$, tem-se

$$\boldsymbol{r}_i^T \boldsymbol{r}_{j+1} = \boldsymbol{r}_i^T \boldsymbol{r}_j - \alpha_i \boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j \longrightarrow \alpha_i \boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j = \boldsymbol{r}_i^T \boldsymbol{r}_j - \boldsymbol{r}_i^T \boldsymbol{r}_{j+1}, \tag{2.46}$$

e com isso, resulta-se em

$$\boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j = \begin{cases} \frac{1}{\alpha_i}\boldsymbol{r}_i^T \boldsymbol{r}_i, \ se\ i = j; \\ -\frac{1}{\alpha_i}\boldsymbol{r}_i^T \boldsymbol{r}_i, \ se\ i = j+1; \\ 0, \ demais\ casos. \end{cases} \tag{2.47}$$

Desta forma, o termo β_{ik} é dado por

$$\beta_{ik} = \begin{cases} \frac{1}{\alpha_{i-1}} \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_{i-1}^T \boldsymbol{A}\boldsymbol{d}_{i-1}}, \ se\ i = j+1; \\ 0, \ se\ i > j+1. \end{cases} \tag{2.48}$$

Considerando $\beta_{i(i-1)} = \beta_i$ e substituindo a Equação (2.36) na Equação (2.48) é possível simplificar a expressão para o termo β_{ik} como dado em

$$\beta_i = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_{i-1}^T \boldsymbol{r}_{i-1}} \longrightarrow \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{r}_{i-1}^T \boldsymbol{r}_{i-1}}. \tag{2.49}$$

Nota-se que muitos dos termos β_{ik} são anulados, desta forma não é necessário o armazenamento na memória de uma quantidade considerável de direções de busca antigas para garantir a $\boldsymbol{A}$-ortogonalidade de novos vetores de descida. Este fato ilustra o grande avanço do método dos gradientes conjugados como algoritmo, uma vez que, a complexidade temporal e espacial são reduzidos de $\mathcal{O}(n^2)$ para $\mathcal{O}(m)$ onde m é o número de entradas não nulas de $\boldsymbol{A}$.

Para resumir, o método dos gradientes conjugados pode ser dado pelos seguintes passos:

1. Inicie $\boldsymbol{u}_0$ aleatoriamente, em seguida, calcule o valor do resíduo inicial $\boldsymbol{r}_0 = \boldsymbol{v} - \boldsymbol{A}\boldsymbol{u}_0$;
2. Determine o valor do passo ótimo para i-ésima iteração utilizando $\alpha_i = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_i}$;
3. Atualizar a solução por meio de $\boldsymbol{u}_{i+1} = \boldsymbol{u} + \alpha_i \boldsymbol{d}_i$;
4. Atualizar o resíduo com $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{A} \boldsymbol{d}_i$;
5. Calcular o multiplicador β_{i+1} do processo de conjugação de Gram-Schmidt utilizando $\beta_{i+1} = \frac{\boldsymbol{r}_{i+1}^T \boldsymbol{r}_{i+1}}{\boldsymbol{r}_i^T \boldsymbol{r}_i}$;
6. Por fim, calcular a nova direção de busca com base no processo de Gram-Schmidt como dado por $\boldsymbol{d}_{i+1} = \boldsymbol{r}_{i+1} + \beta_{i+1} \boldsymbol{d}_i$.

2.5.3 LSSVM Treinado via Gradientes Conjugados

No problema de ajuste do modelo LSSVM, outra maneira de representar o sistema linear KKT é dada por

$$\left[\begin{array}{c|c} 0 & \boldsymbol{y}^T \\ \hline \boldsymbol{y} & \boldsymbol{H} \end{array}\right] \begin{bmatrix} \varepsilon_1 \\ \boldsymbol{\varepsilon}_2 \end{bmatrix} = \begin{bmatrix} d_1 \\ \boldsymbol{d}_2 \end{bmatrix}. \tag{2.50}$$

De tal maneira que $\boldsymbol{H} = \boldsymbol{K} + \gamma^{-1}\boldsymbol{I}$, $\varepsilon_1 = b$, $\boldsymbol{\varepsilon}_2 = \boldsymbol{\alpha}$, $d_1 = 0$ e $\boldsymbol{d}_2 = \boldsymbol{1}$. Esta formulação alternativa permite que o sistema possa ser transformado para

$$\left[\begin{array}{c|c} \boldsymbol{S} & 0 \\ \hline 0 & \boldsymbol{H} \end{array}\right] \begin{bmatrix} \varepsilon_1 \\ \boldsymbol{\varepsilon}_2 + \boldsymbol{H}^{-1}\boldsymbol{y}\varepsilon_1 \end{bmatrix} = \begin{bmatrix} -d_1 + \boldsymbol{y}^T \boldsymbol{H}^{-1} \boldsymbol{d}_2 \\ \boldsymbol{d}_2 \end{bmatrix} \longrightarrow \mathscr{A}\boldsymbol{x} = \mathscr{B}, \tag{2.51}$$

em que $\boldsymbol{S} = \boldsymbol{y}^T \boldsymbol{H}^{-1} \boldsymbol{y} > 0$ é positiva e $\boldsymbol{H}$ é simétrica definida positiva. Isso permite o uso de métodos iterativos eficientes para sua solução numérica, como o método dos gradientes

conjugados (*conjugate gradient*, CG) de Hesteness-Stiefel (Suykens e Vandewalle, 1999), que apresenta uma complexidade intermediária entre métodos de primeira ordem (como o gradiente descendente) e métodos de segunda ordem (como os de Newton ou quase Newton), além de evitar as complexidades numéricas relacionadas à inversão de matrizes. O procedimento de obtenção da solução estimada para a LSSVM baseado no método CG de Hestenes-Stiefel pode ser resumido conforme o Algoritmo 1 (Suykens e Vandewalle, 1999).

Algoritmo 1: LSSVM via Hesteness-Stiefel CG

Entrada: $\boldsymbol{x}_0, i = 0, \boldsymbol{r}_0 = \mathscr{B}$

início
  while $\boldsymbol{r}_i \neq \mathbf{0}$ **do**
    $i = i + 1$
    if $i = 1$ **then**
      $\boldsymbol{p}_1 = \boldsymbol{r}_0$
      else
        $\beta_i = \frac{\boldsymbol{r}_{i-1}^T \boldsymbol{r}_{i-1}}{\boldsymbol{r}_{i-2}^T \boldsymbol{r}_{i-2}}$
        $\boldsymbol{p}_i = \boldsymbol{r}_{i-1} + \beta_i \boldsymbol{p}_{i-1}$
      end
    end
    $\lambda_i = \frac{\boldsymbol{r}_{i-1}^T \boldsymbol{r}_{i-1}}{\boldsymbol{p}_i^T \mathscr{A} \boldsymbol{p}_i}$
    $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \lambda_i \boldsymbol{p}_i$
    $\boldsymbol{r}_i = \boldsymbol{r}_{i-1} - \lambda_i \mathscr{A} \boldsymbol{p}_i$
  end
  $\boldsymbol{x} = \boldsymbol{x}_i$
fim

Resultado: Solução aproximada do sistema $\mathscr{A}\boldsymbol{x} = \mathscr{B}$

2.6 Solução Baseada no Método de Levenberg-Marquardt

Esta seção atenta-se a discutir outra forma de resolução do problema de otimização da LSSVM, apresentada inicialmente em Neto e Barreto (2009). O método LM, é um algoritmo de otimização iterativo que aplica treinamento em lote, no qual consiste em aperfeiçoamento do método Gauss-Newton e busca minimizar o erro quadrático como dado a seguir (Moré, 2006)

$$\boldsymbol{L}(\boldsymbol{z}) = \frac{1}{2}\sum_{i=1}^{N} e_i^2(\boldsymbol{z}) = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e}. \tag{2.52}$$

A ideia básica do método LM é combinar o método do gradiente descendente (1° ordem) com o método de Newton (2° ordem) para ajustar uma função não linear aos dados

observados, minimizando a soma dos quadrados das diferenças entre os valores previstos pela função e os valores observados.

Dada a função custo $\boldsymbol{L}(\boldsymbol{z})$ com base nos erros quadráticos, mostrada na Equação (2.52), na qual $e_i(\boldsymbol{z})$ caracteriza o erro da i-ésima amostra e $\boldsymbol{e}(\boldsymbol{z})$ é o vetor de erros e utilizando a aproximação por série de Taylor em torno de um ponto z_0 para o gradiente $\nabla\boldsymbol{L}(\boldsymbol{z})$, chega-se a seguinte formulação

$$\nabla\boldsymbol{L}(\boldsymbol{z}) = \nabla\boldsymbol{L}(\boldsymbol{z_0}) + (\boldsymbol{z}-\boldsymbol{z}_0)^T\nabla^2\boldsymbol{L}(\boldsymbol{z}_0) + \ldots \tag{2.53}$$

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

$$\nabla\boldsymbol{L}(\boldsymbol{z}) = 0. \tag{2.54}$$

Considerando apenas até os termos de segunda ordem na expansão de Taylor, pode-se aproximar $\Delta\boldsymbol{z}$ pela seguinte expressão

$$\Delta\boldsymbol{z} = -[\nabla^2\boldsymbol{L}(\boldsymbol{z}_0)]^{-1}\nabla\boldsymbol{L}(\boldsymbol{z}_0), \tag{2.55}$$

em que o gradiente $\nabla\boldsymbol{L}(\boldsymbol{z})$ e a Hessiana $\nabla^2\boldsymbol{L}(\boldsymbol{z})$ podem ser escritos, respectivamente, com base na matriz jacobiana $\boldsymbol{J}(\boldsymbol{z})$, como

$$\nabla\boldsymbol{L}(\boldsymbol{z}) = \boldsymbol{J}(\boldsymbol{z})\boldsymbol{e}(\boldsymbol{z}), \tag{2.56}$$

e

$$\nabla^2\boldsymbol{L}(\boldsymbol{z}) = \boldsymbol{J}^T(\boldsymbol{z})\boldsymbol{J}(\boldsymbol{z}) + \sum_{i=1}^{N} e_i(\boldsymbol{z})\nabla^2 e_i(\boldsymbol{z}). \tag{2.57}$$

Por conta da complexidade de se calcular a matriz Hessiana, foram propostos métodos que aplicam aproximações, como é o caso do método de Levenberg-Marquardt. Levando em conta que o mesmo é um método Quasi-Newton, assim como o método de Gauss-Newton, assume-se que

$$\sum_{i=1}^{N} e_i(\boldsymbol{z})\nabla^2 e_i(\boldsymbol{z}) \approx 0. \tag{2.58}$$

Do exposto, o termo adicional mostrado na Equação (2.57), para o cálculo da Hessiana aproximada pode ser retirado e neste contexto a variação pode ser dada por

$$\Delta z = -[\boldsymbol{J}^T(z)\boldsymbol{J}(z)]^{-1}\boldsymbol{J}(z)\boldsymbol{e}(z). \tag{2.59}$$

Considerando que a matriz Hessiana aproximada pode não possuir inversa, um procedimento de regularização de Tikhonov (Golub *et al.*, 1999) pode ser aplicado, fazendo com que a variação e a regra de atualização resultantes sejam dadas por

$$\Delta z = -[\boldsymbol{J}^T(z)\boldsymbol{J}(z) + \mu I]^{-1}\boldsymbol{J}(z)\boldsymbol{e}(z), \tag{2.60}$$

e

$$z_{t+1} = z_t + \Delta z. \tag{2.61}$$

Após os trabalhos de Levenberg (1944), Marquardt (1963) foi proposto uma substituição da matriz identidade pela matriz diagonal da Hessiana aproximada, para contornar outro problema, o crescimento acelerado do μ, uma vez que quando isso acontece a informação dada pela Hessiana aproximada não é útil no cálculo da atualização. Esta alteração resulta na seguinte expressão para a variação da variável dual

$$\Delta z = -[\boldsymbol{J}^T(z)\boldsymbol{J}(z) + \mu \ \ diag(\boldsymbol{J}^T(z)\boldsymbol{J}(z))]^{-1}\boldsymbol{J}(z)\boldsymbol{e}(z). \tag{2.62}$$

Dada a Equação (2.62), e sabendo que o método LSSVM é um problema de minimização em que se deseja solucionar o sistema linear, $A\boldsymbol{z} = \boldsymbol{v}$, pode-se calcular a matriz jacobiana com base nas derivadas parciais da função erro, ou seja

$$\boldsymbol{J}(z) = \frac{\partial \boldsymbol{e}(z)}{\partial \boldsymbol{z}} = -\boldsymbol{A}, \tag{2.63}$$

em que

$$\boldsymbol{e}(z) = \boldsymbol{v} - \boldsymbol{A}z. \tag{2.64}$$

Por fim, a atualização do vetor $\boldsymbol{z}$ que contém os multiplicadores de Lagrange e o viés pode ser determinada pela expressão a seguir

$$z_{i+1} = z_i + [\boldsymbol{A}^T\boldsymbol{A} + \mu \ \ diag(\boldsymbol{A}^T\boldsymbol{A})]^{-1}\boldsymbol{A}^T\boldsymbol{e}(z), \tag{2.65}$$

em que i corresponde a i-ésima iteração e, como dito, o componente $\mu \ diag(\boldsymbol{A}^T\boldsymbol{A})$ representa um termo de regularização.

Neste capitulo, foram apresentadas as formulações para o problema primal e dual do treinamento de LSSVMs, tanto nos contextos de classificação binária como para problemas de aproximação de funções. Além disso, as metodologias mais comumente empregadas na solução do problema primal também são descritas, destacando-se o uso dos algoritmos CG de Hesteness-Stiefel e de Levenberg-Marquardt. No próximo capítulo, serão consideradas as principais metodologias para obtenção de LSSVMs esparsas baseadas em poda iterativa, bem como, são apresentadas algumas abordagens baseadas no algoritmo SMO para a solução do problema dual no ajuste de LSSVMs.

3 VARIANTES ESPARSAS E ALGORITMO SMO

Neste capítulo, aborda-se as principais e mais comumente metodologias empregadas para a esparsificação do vetor de multiplicadores de Lagrange no modelo LSSVM. No caso, apresentam-se as variantes P-LSSVM, IP-LSSVM e o modelo Levenberg-Marquardt com poda iterativa. Em seguida, são apresentados os algoritmos SMO de primeira e segunda ordem e o método CSMO com estratégia FGWSS para a seleção do par de multiplicadores de Lagrange que passarão por atualização dentro da iteração, tal par é denominado conjunto de trabalho.

Em métodos de poda, a solução esparsa é obtida através da remoção dos vetores-suporte induzindo a eliminação de colunas da matriz $\boldsymbol{A}$ o que equivale a zerar os multiplicadores de Lagrange associados (Carvalho e Braga, 2009). Durante a poda, um determinado método remove um número fixo de colunas de $\boldsymbol{A}$ e suas linhas correspondentes em $\boldsymbol{z}$, preservando assim as dimensões corretamente.

3.1 Métodos de Poda Clássicos

3.1.1 P-LSSVM

O modelo P-LSSVM foi proposto em Suykens *et al.* (2000) como uma alternativa ao problema de carência de esparsidade das soluções no treinamento de LSSVMs. A poda dos vetores-suporte, $\boldsymbol{x}_i$, é realizada de acordo com o valor absoluto dos multiplicadores Lagrange associados, $|\alpha_i|$, de tal forma que os vetores que correspondem aos menores multiplicadores de Lagrange em valor absoluto são eliminados a cada iteração do modelo (Suykens e Vandewalle, 1999).

A desvantagem desse procedimento está no fato de que uma vez que a poda tenha sido realizada em uma determinada iteração, a LSSVM deve ser novamente ajustada para o novo conjunto de dados atualizados sem os vetores removidos, o que o torna pouco eficiente em termos de processamento (Zeng e Chen, 2005).

Além disso, o critério de parada deste método é o desempenho do classificador. Para isso, utiliza-se um conjunto de validação, que é um subconjunto de treinamento, e a cada iteração é verificado se o desempenho continua o mesmo com base nesse conjunto. Para as simulações deste trabalho, aplicou-se como critério de parada a porcentagem de redução. De forma resumida, o método P-LSSVM tem quatro passos principais, como apresentado no Algoritmo 2.

Algoritmo 2: P-LSSVM

Entrada: $\boldsymbol{X}, \boldsymbol{y}$

1 **início**

2 Treina o LSSVM considerando o conjunto de treinamento completo.

3 Remove um pequeno número de padrões de treinamento, aqueles com menores valores de $|\alpha_i|$.

4 Retreina o LSSVM utilizando o método da pseudo-inversa com o conjunto de treino reduzido.

5 Retorna ao PASSO 2 até que ocorra a estabilização do algoritmo onde o desempenho sobre um conjunto de validação varia abaixo de um determinado limiar. Os vetores-suporte serão aqueles do conjunto reduzido da iteração anterior.

6 **fim**

Resultado: Solução ótima esparsa $\boldsymbol{u}^*$

3.1.2 IP-LSSVM

O método IP-LSSVM proposto em Carvalho e Braga (2009) é uma alternativa ao P-LSSVM na tarefa de esparsificar o vetor de multiplicadores de Lagrange ótimos no treinamento da LSSVM. O critério adotado é de que os padrões mais próximos da superfície de decisão são os melhores candidatos a vetores-suporte para o problema de treinamento de LSSVMs, o que é um racional similar ao aplicado na detecção dos vetores-suporte para o modelo SVM.

De fato, este método é bem similar ao P-LSSVM com a diferença de que a poda dos padrões é realizada com base no valores diretos dos multiplicadores de Lagrange, ou seja α_i, e não nos seus valores absolutos como o P-LSSVM (Carvalho e Braga, 2009).

Esse método obtém o conjunto de vetores-suporte em dois passos principais. No primeiro, aplica-se a inversa para encontrar a solução do sistema linear. Posteriormente, na segunda, é empregado o método da pseudo inversa para resolver o novo sistema linear modificado no processo de poda, no qual, teve colunas removidas com base nos multiplicadores de Lagrange obtidos na etapa anterior. O processo de treinamento do classificador IP-LSSVM é descrito pelo Algoritmo 3.

3.1.3 Poda Iterativa com Levenberg-Marquardt

A poda iterativa no classificador Levenberg-Marquardt LSSVM de tamanho fixo (*fixed sized Levenberg-Marquardt LSSVM*, FSLM-LSSVM) é um método derivado do LSSVM tradicional e apresentado originalmente em Leao e Neto (2022), que visa encontrar o ponto ótimo z^* do sistema linear KKT de forma iterativa, removendo uma quantidade fixa de vetores-suporte.

Algoritmo 3: IP-LSSVM

Entrada: $\boldsymbol{X}, y, \tau$

início

Treina o LSSVM considerando o conjunto de treinamento completo, utilizando o método da inversa, uma vez que, $\boldsymbol{A}$ é uma matriz quadrada.

Especifique o valor do parâmetro τ, que define a porcentagem de vetores de treino que serão vetores-suporte.

Ordene os padrões de treino $(\boldsymbol{x}_i, d_i)_{i=1}^{N=1}$ com base em seus valores de α_i.

Seleciona uma porcentagem de padrões de treino $(1-\omega)$ que correspondem aos menores valores de α_i.

Construa a matriz não quadrada $\boldsymbol{A}_{rs}$ pela remoção das colunas de $\boldsymbol{A}$ associadas com os vetores com menores valores de α_i.

Resolver o sistema linear $\boldsymbol{u}_{rs} = \boldsymbol{A}^* v$, com $\boldsymbol{A}^* = (\boldsymbol{A}_{rs}^T \boldsymbol{A}_{rs})^{-1} \boldsymbol{A}_{rs}^T$

Os vetores-suportes serão aqueles relacionados a colunas da matriz $\boldsymbol{A}_{rs}$.

Os valores de α_i e b podem ser obtidos $\boldsymbol{u}_{rs}$, enquanto os valores de α_i para os demais padrões são iguais a zero.

fim

Resultado: Solução ótima esparsa $\boldsymbol{u}^*$

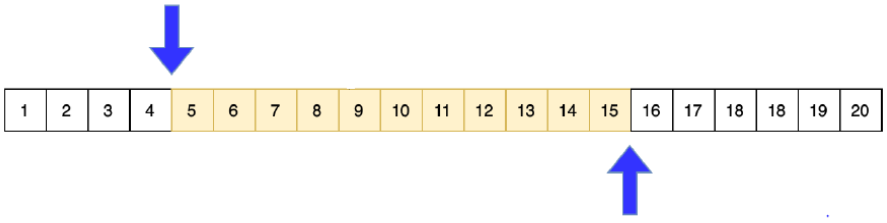
Para isso, o método de Levenberg-Marquardt foi adaptado.

3.1.3.1 Janela de Poda

Considerando a metodologia proposta, deve-se definir quando o algoritmo deve reduzir o número de vetores-suporte. Com isso em mente, o método da janela de poda foi estabelecido, indicando as iterações em que os vetores-suporte serão removidos durante a execução do modelo.

Portanto, dado um número máximo de iterações N_{max}, os padrões serão podados no intervalo $\kappa \leq t \leq N_{max} - \kappa$. Por exemplo, na Figura 4, foram definidos $N_{max} = 20$ e $\kappa = 5$ Assim, a janela de poda será $5 \leq t \leq 15$. Vale destacar que, nas simulações computacionais realizadas neste trabalho, κ foi definido como igual a 5.

Figura 4 – Janela de poda com $\kappa = 5$



Fonte: Elaborada pelo autor.

3.1.3.2 *Soluções esparsas*

Para o método FSLM-LSSVM, o número de padrões podados (colunas removidas de $\boldsymbol{A}$) em cada iteração da janela é dado por

$$\tau = \frac{N_{train} - N_{train}R}{N_{max} - 2\kappa}, \tag{3.1}$$

em que:

- N_{train}: número total de exemplos do conjunto de treinamento;
- R: porcentagem de redução;
- N_{max}: número total de iterações; e
- κ: define a janela de poda.

Caso τ não seja um número inteiro, na última iteração da janela são podados $\tau + \Psi$, onde Ψ é simplesmente o resto da divisão dado por[1]

$$\Psi = \{N_{train} - N_{train}R\} \bmod \{N_{max} - 2\kappa\}. \tag{3.2}$$

O Algoritmo 4 descreve o fluxo do FSLM-LSSVM, recebendo como entrada: coeficiente μ, porcentagem de redução R, número de iterações N_{max} e intervalo κ.

3.2 Metodologias Baseadas no Algoritmo SMO

O treinamento de LSSVMs padrão, P-LSSVM e IP-LSSVM consiste na solução do problema de otimização primal como dado pela Equação (2.1), no entanto, ainda é possível explorar o problema dual correspondente, como é feito nas metodologias para treinamento de LSSVMs baseadas no algoritmo SMO (López e Suykens, 2011; Yu *et al.*, 2023b; Yu *et al.*, 2023a).

3.2.1 SMO de Primeira Ordem

O algoritmo SMO resolve o problema dual das LSSVMs, que é um programa quadrático, pela decomposição do problema geral em sub-problemas com a menor complexidade

[1] O operador a mod b indica o operador módulo, ou seja, o resto da divisão de a por b

Algoritmo 4: Algoritmo FSLM-LSSVM

Entrada: $\mu, R, N_{max}, \kappa, \varepsilon$

início
- Inicialização aleatória para $\boldsymbol{z}^0$ e $K = 0$.
- **while** $K \leq N_{max}$ *OR* $\Delta MSE > \varepsilon$ **do**
  - Construir a matriz $\boldsymbol{A}$ e o vetor $\boldsymbol{v}$, baseado no conjunto de treinamento completo.
  - Calcule o erro, $\boldsymbol{e}_i(\boldsymbol{z}_i) = \boldsymbol{b}\boldsymbol{A}^T\boldsymbol{z}_i$, da i-ésima iteração.
  - Atualiza o valor de $\boldsymbol{z}$ usando a Equação 2.65.
  - **if** $\Delta MSE < 0$ **then**
    - Continue atualizando $\boldsymbol{z}_i$ usando a Equação 2.65.
    - **else**
      - Faça $\mu = \frac{\mu}{10}$.
    - **end**
  - **end**
  - **if** $\kappa < K < N_{max} - \kappa$ **then**
    - Remova τ colunas de $\boldsymbol{A}$ e linhas de $\boldsymbol{z}$ que apresentam os menores valores de $|\alpha_i|$.
  - **end**
  - **if** $K = N_{max} - \kappa$ **then**
    - Remova $\tau + \Psi$ colunas de $\boldsymbol{A}$ e linhas de $\boldsymbol{z}$ que apresentam os menores valores de $|\alpha_i|$.
  - **end**
  - $K = K + 1$
- **end**

fim

Resultado: Solução ótima esparsa $\boldsymbol{u}^*$

possível, realizando a atualização de dois multiplicadores de Lagrange por vez. Essa simplificação permite que os sub-programas possam apresentar solução analítica fechada, resultando em uma metodologia com complexidade linear e de fácil implementação (Platt, 1998).

Retornando ao problema dual mostrado na Equação (2.26), nota-se que ao se aplicar as condições de otimalidade para este problema convexo, resulta-se na condição de gradiente nulo, como indicado abaixo

$$\nabla\mathscr{D}(\boldsymbol{\alpha}) = \boldsymbol{g}(\boldsymbol{\alpha}) = \tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y} \longrightarrow g_i(\boldsymbol{\alpha}) = \sum_{j=1}^{N} \alpha_j \tilde{K}_{ij} - y_i. \tag{3.3}$$

Perceba que para um $\boldsymbol{\alpha}$ que satisfaça a condição $max_i(g_i(\boldsymbol{\alpha})) = min_i(g_i(\boldsymbol{\alpha}))$, tem-se que o mesmo é a solução para o problema dual mostrado na Equação (2.26). Este fato fornece a base para a definição da estratégia de seleção do conjunto de trabalho para o SMO de primeira ordem e segunda ordem, bem como, para o critério de parada do método.

Considere o (α_i^k, α_j^k) o par de multiplicadores de Lagrange selecionados na k-ésima iteração. As atualizações deste multiplicadores são dadas pelas expressões que se seguem

$$\begin{aligned} \alpha_i^{k+1} &\Leftarrow \alpha_i^k + \Delta\alpha_i, \\ \alpha_j^{k+1} &\Leftarrow \alpha_j^k + \Delta\alpha_j, \\ \alpha_l^{k+1} &\Leftarrow \alpha_l^k \quad \forall l \neq i, j. \end{aligned} \tag{3.4}$$

Além disso, ao avaliar o ganho funcional obtido a cada iteração pela atualização dos dois multiplicadores de Lagrange do conjunto de trabalho, tem-se que o mesmo é dado por

$$\begin{aligned} f_G(\Delta\boldsymbol{\alpha}^k) &= \mathscr{D}(\boldsymbol{\alpha}^k) - \mathscr{D}(\boldsymbol{\alpha}^{k+1}), \\ f_G(\Delta\boldsymbol{\alpha}^k) &= (\frac{1}{2}\boldsymbol{\alpha}^{\boldsymbol{k}T}\tilde{\boldsymbol{K}}\boldsymbol{\alpha}^{\boldsymbol{k}} - \boldsymbol{y}^T\boldsymbol{\alpha}^{\boldsymbol{k}}) - (\frac{1}{2}\boldsymbol{\alpha}^{\boldsymbol{k+1}T}\tilde{\boldsymbol{K}}\boldsymbol{\alpha}^{\boldsymbol{k+1}} - \boldsymbol{y}^T\boldsymbol{\alpha}^{\boldsymbol{k+1}}), \\ f_G(\Delta\boldsymbol{\alpha}^k) &= -\frac{1}{2}\begin{bmatrix} -\Delta\alpha_j \\ \Delta\alpha_j \end{bmatrix}^T \begin{bmatrix} \tilde{K}_{ii} & \tilde{K}_{ij} \\ \tilde{K}_{ji} & \tilde{K}_{jj} \end{bmatrix} \begin{bmatrix} -\Delta\alpha_j \\ \Delta\alpha_j \end{bmatrix} + \begin{bmatrix} g_i(\boldsymbol{\alpha}^{\boldsymbol{k}}) \\ g_j(\boldsymbol{\alpha}^{\boldsymbol{k}}) \end{bmatrix}^T \begin{bmatrix} -\Delta\alpha_j \\ \Delta\alpha_j \end{bmatrix}. \end{aligned} \tag{3.5}$$

Por ser uma função quadrática, é possível determinar o incremento ótimo, $\Delta\boldsymbol{\alpha}^k$, que maximiza o ganho funcional, através da condição de gradiente nulo, $f'(\Delta\boldsymbol{\alpha}^k) = 0$, com isso o incremento ótimo é dado por

$$\Delta\alpha_j = \frac{g_j(\boldsymbol{\alpha}^k) - g_i(\boldsymbol{\alpha}^k)}{\tilde{K}_{ii} + \tilde{K}_{jj} - \tilde{K}_{ij} - \tilde{K}_{ji}}. \tag{3.6}$$

Substituindo este resultado na Equação (3.5), resulta no ganho funcional maximizado indicado pela formulação que se segue

$$f_G(\Delta\alpha_j) = \frac{(g_j(\boldsymbol{\alpha}^k) - g_i(\boldsymbol{\alpha}^k))^2}{2(\tilde{K}_{ii} + \tilde{K}_{jj} - \tilde{K}_{ij} - \tilde{K}_{ji})}. \tag{3.7}$$

Portanto, a estratégia de seleção do conjunto de trabalho pode ser dada pela seleção do par de índices (i, j) que maximiza o ganho funcional, isto é

$$(i, j) = \arg\max_{m,l} \left[\frac{(g_l(\boldsymbol{\alpha}^k) - g_m(\boldsymbol{\alpha}^k))^2}{2(\tilde{K}_{mm} + \tilde{K}_{ll} - \tilde{K}_{ml} - \tilde{K}_{lm})} \right]. \tag{3.8}$$

No algoritmo SMO de primeira ordem, o denominador da Equação (3.8) é negligenciado devido alto requerimento para o seu cômputo. A avaliação do denominador tornaria a

complexidade desta metodologia quadrática (Yu *et al.*, 2023b). Neste cenário, a heurística de seleção do par de multiplicadores é como se segue

$$(i,j) = \arg\max_{m,l} (g_l(\boldsymbol{\alpha}^k) - g_m(\boldsymbol{\alpha}^k))^2. \tag{3.9}$$

Obviamente, o mesmo resultado dado pela Equação (3.9) pode ser obtido pela expressão

$$i = \arg\min_{l} g_l(\boldsymbol{\alpha}^k), \quad j = \arg\max_{m} g_m(\boldsymbol{\alpha}^k). \tag{3.10}$$

O detalhamento do procedimento de cálculo para o SMO de primeira ordem é mostrado no Algoritmo 5, onde $\boldsymbol{g}(\boldsymbol{\alpha}^k) = [g_1(\boldsymbol{\alpha}^k), \ldots, g_n(\boldsymbol{\alpha}^k)]^T$ indica o vetor gradiente de $\mathscr{D}(\boldsymbol{\alpha})$.

Algoritmo 5: SMO de primeira ordem

Entrada: $\boldsymbol{X}$, Limiar ε

início

  Inicialize $\alpha^0 = \mathbf{0}$, $\boldsymbol{g}(\alpha^0) = -\boldsymbol{y}$ e $k = 0$.

  while $max_i(\boldsymbol{g}_i(\boldsymbol{\alpha})) - min_i(\boldsymbol{g}_i(\boldsymbol{\alpha})) > \varepsilon$ **do**

    Selecione o conjunto de trabalho (i,j) utilizando a Equação (3.9); Calcule $\Delta\boldsymbol{\alpha}^k$ com a Equação (3.6); Atualize a variável dual $\boldsymbol{\alpha}^{k+1}$ utilizando a Equação (3.4); Atualize o gradiente $\boldsymbol{g}(\boldsymbol{\alpha}^k)$; Faça $k \longleftarrow k+1$ e retorne ao PASSO 2;

  end

fim

Resultado: Solução Dual $\boldsymbol{\alpha}^*$

3.2.2 SMO de Segunda Ordem

A grande diferença do SMO de segunda ordem está no fato de que na seleção do conjunto de trabalho, considera-se o denominador apresentado na Equação (3.8). Ao considerar este termo na avaliação e seleção do par de multiplicadores de Lagrange, é necessário uma busca que torna a complexidade de processamento quadrática, o que pode inviabilizar o uso de tal metodologia para grandes bases de dados (López e Suykens, 2011). Para tornar viável o uso do SMO de segunda ordem, Platt (1998), López e Suykens (2011) empregaram um método que primeiro utiliza o SMO de primeira ordem para selecionar o índice i e com este determina o índice j utilizando o SMO de segunda ordem, como dado em

$$
\begin{aligned}
i &= \arg\min_{l} g_l(\boldsymbol{\alpha}_k), \\
j &= \arg\max_{l \neq i} \left[\frac{(g_l(\boldsymbol{\alpha}^k) - g_i(\boldsymbol{\alpha}^k))^2}{2(\tilde{K}_{ii} + \tilde{K}_{ll} - \tilde{K}_{il} - \tilde{K}_{li})} \right].
\end{aligned}
\tag{3.11}
$$

Embora o algoritmo SMO de segunda ordem aumente o número de operações para manipulação da matriz de *kernel*, o ganho funcional é maior do que aquele proporcionado pelo SMO de primeira ordem a cada iteração. O procedimento de cálculo para esta metodologia é equivalente ao de primeira ordem, como descrito no Algoritmo 5, apenas atualizando o passo 4 para acomodar a nova estratégia de seleção do conjunto de trabalho.

3.2.3 Seleção do Conjunto de Trabalho pelo Ganho Funcional (FGWSS)

O SMO de segunda ordem apresenta uma desvantagem, relacionada a seleção do índice i, uma vez que $g_i(\boldsymbol{\alpha})$ pode não satisfazer a condição KKT, de tal forma que pode existir um índice $\hat{i}$ que satisfaz $|g_i(\boldsymbol{\alpha})| \leq g_{\hat{i}}(\boldsymbol{\alpha})$. Em Fan *et al.* (2005) foi proposto a seleção do índice i baseado no valor máximo de $|\boldsymbol{g}(\boldsymbol{\alpha})|$ e, em seguida, a seleção de j utilizando o SMO de segunda ordem, ou seja, i é obtido por

$$
i = \arg\max_{l} |g_l(\boldsymbol{\alpha}^k)|, \tag{3.12}
$$

enquanto j é dado por

$$
j = \arg\max_{l \neq i} \left[\frac{(g_l(\boldsymbol{\alpha}^k) - g_i(\boldsymbol{\alpha}^k))^2}{2(\tilde{K}_{ii} + \tilde{K}_{ll} - \tilde{K}_{il} - \tilde{K}_{li})} \right]. \tag{3.13}
$$

No caso do mesmo gradiente $\boldsymbol{g}(\boldsymbol{\alpha})$, usar o conjunto de trabalho (i, j) selecionado pelo FGWSS sempre garante que a variação na função dual seja maior ou igual a estratégia de máxima violação de pares (*maximum violation pair*, MVP) (Yu *et al.*, 2023b). O procedimento de cálculo utilizando a estratégia FGWSS do SMO é equivalente ao de primeira ordem, conforme descrito no Algoritmo 4, subsituindo apenas o passo 4 pelo novo esquema de seleção do conjunto de trabalho.

3.2.4 SMO Conjugado de Ganho Funcional LSSVM (CSMO-LSSVM)

A principal diferença do SMO Conjugado de Ganho Funcional LSSVM (*conjugate functional gain SMO LSSVM*, CSMO-LSSVM) para a abordagem do SMO de primeira e segunda

ordem, está no uso da heurística FGWSS para a seleção do conjunto de trabalho, bem como, na direção de atualização da variável dual, que é dada por

$$\begin{cases} \boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k + \rho_k \boldsymbol{z}_k, \\ \boldsymbol{z}_k = \boldsymbol{h}_{ij}^k + r_k \boldsymbol{z}_{k-1}. \end{cases} \tag{3.14}$$

A direção $\boldsymbol{h}_{ij}^k$ é determinada pelo FGWSS, e r_k é o parâmetro conjugado. O tamanho do passo ótimo pode ser obtido através do critério de busca linear exata [2] (Pérez e Prudente, 2019), considerando a função $\phi(\rho_k) = \mathscr{D}(\boldsymbol{\alpha}^k + \rho_k \boldsymbol{z}_k)$ e aplicando a condição de gradiente nulo no ponto ótimo como detalhado a seguir

$$\begin{aligned} \phi'(\rho_k) &= \boldsymbol{z}_k^T \nabla \mathscr{D}(\boldsymbol{\alpha}^k + \rho_k \boldsymbol{z}_k) \\ &= \boldsymbol{z}_k^T \boldsymbol{g}(\boldsymbol{\alpha}^k + \rho_k \boldsymbol{z}_k) \\ &= \boldsymbol{z}_k^T [\tilde{\boldsymbol{K}}(\boldsymbol{\alpha}^k + \rho_k \boldsymbol{z}_k) - \boldsymbol{y}] \\ &= \boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{\alpha}^k + \rho_k \boldsymbol{z}_k^T \tilde{K} \boldsymbol{z}_k - \boldsymbol{z}_k^T \boldsymbol{y} \\ &= \boldsymbol{z}_k^T (\tilde{\boldsymbol{K}} \boldsymbol{\alpha}^k - \boldsymbol{y}) + \rho_k \boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k - \boldsymbol{z}_k^T \boldsymbol{y} \\ &= \boldsymbol{z}_k^T \boldsymbol{g}(\boldsymbol{\alpha}^k) + \rho_k \boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k, \end{aligned} \tag{3.15}$$

assim, fazendo $\phi'(\rho_k) = 0$, pode-se obter o tamanho do passo ótimo como dado abaixo

$$\rho_k^* = -\frac{\boldsymbol{z}_k^T \boldsymbol{g}(\alpha^k)}{\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k}. \tag{3.16}$$

Além disso, o ganho funcional desta metodologia pode ser expresso como segue

$$f_G = \mathscr{D}(\boldsymbol{\alpha}^k) - \mathscr{D}(\boldsymbol{\alpha}^{k+1}) = -\frac{1}{2}\rho \boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k - \rho \boldsymbol{z}_k^T (\tilde{\boldsymbol{K}} \boldsymbol{\alpha}^k - \boldsymbol{y}). \tag{3.17}$$

Substituindo a Equação (3.16) na Equação (3.17) e utilizando o fato de que $\boldsymbol{z}_k$ é direção de descida, ou seja $\boldsymbol{z}_{k-1}^T \boldsymbol{g}(\boldsymbol{\alpha}^k) = 0$, obtém-se a expressão

$$f_G = \mathscr{D}(\boldsymbol{\alpha}^k) - \mathscr{D}(\boldsymbol{\alpha}^{k+1}) = \frac{1}{2} \frac{(\boldsymbol{g}(\boldsymbol{\alpha}^k)^T \boldsymbol{h}_{ij}^k)^2}{\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k}, \tag{3.18}$$

[2] Este critério visa determinar o tamanho do passo que fornece o mínimo valor da função objetivo ao longo da direção de busca.

em que,

$$\psi(r) = \boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k = (\boldsymbol{h}_{ij}^k + r\boldsymbol{z}_{k-1})^T \tilde{\boldsymbol{K}} (\boldsymbol{h}_{ij}^k + r\boldsymbol{z}_{k-1}). \tag{3.19}$$

Assim, com intuito de maximizar o ganho funcional por iteração, f_G, determina-se o parâmetro conjugado que minimiza o denominador, $\psi(r)$. Utilizando a condição de gradiente nulo, tem-se que

$$\psi'(r) = 2(\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}^k + r\boldsymbol{z}_{k-1} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}) = 0 \longrightarrow r^* = -\frac{\boldsymbol{z}_{k-1}^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}^k}{\boldsymbol{z}_{k-1}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}}. \tag{3.20}$$

Os detalhes para o método CSMO-LSSVM são apresentados no Algoritmo 6.

Algoritmo 6: CSMO-LSSVM

Entrada: $\boldsymbol{X}$, Limiar ε

início

Inicialize $\alpha^0 = \boldsymbol{0}$, $\boldsymbol{g}(\alpha^0) = -\boldsymbol{y}$, $\boldsymbol{z}_0 = \boldsymbol{t}_0 = \boldsymbol{0}$, $\tau^0 = 1$ e $k = 0$

while $max_i(\boldsymbol{g}_i(\boldsymbol{\alpha})) - min_i(\boldsymbol{g}_i(\boldsymbol{\alpha})) > \varepsilon$ **do**

$k \longleftarrow k+1$.

Selecione o conjunto de trabalho (i_k, j_k) de acordo com as Equações (3.12) e (3.14).

Calcule as colunas da matriz de *kernel* $[\tilde{\boldsymbol{K}}]_{:i_k}$ e $[\tilde{\boldsymbol{K}}]_{:j_k}$.

$r_k \longleftarrow \frac{t_{i_k}^{k-1} - t_{j_k}^{k-1}}{\tau^{k-1}}$

$\boldsymbol{z}_k \longleftarrow \boldsymbol{h}^k + r_k \boldsymbol{z}_{k-1}$

$\boldsymbol{t}^k \longleftarrow [\tilde{\boldsymbol{K}}]_{:i_k} - [\tilde{\boldsymbol{K}}]_{:j_k} + r_k \boldsymbol{t}^{k-1}$

$\tau^k \longleftarrow t_{j_k}^k - t_{i_k}^k$

Calcule $\rho_k \longleftarrow \frac{\boldsymbol{g}_{i_k}(\boldsymbol{\alpha^k}) - \boldsymbol{g}_{j_k}(\boldsymbol{\alpha^k})}{\tau^k}$

Atualize a variável dual $\boldsymbol{\alpha}^k \longleftarrow \boldsymbol{\alpha}^{k-1} + \rho_k \boldsymbol{z}_k$

Atualize o gradiente $\boldsymbol{g}(\boldsymbol{\alpha}^k) \longleftarrow \boldsymbol{g}(\boldsymbol{\alpha}^{k-1}) + \rho_k \boldsymbol{t}^k$

end

fim

Resultado: Solução Dual $\boldsymbol{\alpha}^*$

Neste capítulo, foram apresentados os métodos mais comumente empregados para a tarefa de esparsificação do vetor de multiplicadores de Lagrange ótimos obtidos durante a etapa de treinamento de LSSVMs. Um maior enfoque foi dado aos métodos que utilizam poda iterativa, como o P-LSSVM e IP-LSSVM. Além disso, também foi detalhado o recente método FSLM-LSSVM, que combina o uso do algoritmo LM com um esquema de poda iterativa que favorece a obtenção de uma solução esparsa com rápido treinamento.

A fim de trazer o racional adotado nas novas propostas, foi apresentado o método CSMO-LSSVM que utiliza um algoritmo SMO com direção de descida conjugada para a solução do problema dual correspondente, este serve como base na fundamentação da primeira proposta. O próximo capítulo apresenta todo o formalismo matemático associado às duas novas propostas, considerando suas provas de convergência, bem como, uma estimativa de suas complexidades computacionais. 1

4 MÉTODOS PROPOSTOS

Neste capítulo, dois novos métodos são propostos para o treinamento rápido e esparso do modelo LSSVM. Na primeira proposta, o algoritmo SMO de três termos conjugados, originalmente aplicado para o treinamento de SVMs (Yu *et al.*, 2023a), foi adaptado para a solução do problema dual decorrente do treinamento de LSSVMs com a adição de um método de poda iterativa baseada no ganho funcional do problema dual (Zeng e Chen, 2005).

Enquanto na segunda proposta, o problema dual é resolvido via o uso de um recentemente publicado método do gradiente conjugado espectral (Wang *et al.*, 2020), que utiliza um esquema de escolha dos parâmetros espectral e conjugado que favorece o desenvolvimento de uma nova direção de busca que satisfaz a propriedade espectral e a condição de descida simultaneamente, favorecendo alto ganho funcional a cada iteração. Além disso, soluções esparsas são obtidas pelo uso de um novo método de poda iterativa de tamanho variável, que se baseia na proximidade do padrão ao hiperplano de decisão para a realização da poda.

4.1 SMO Conjugado de Três Termos LSSVM (Proposta 1)

O sucesso de aplicações envolvendo o método das direções conjugadas tem impulsionado diversas pesquisas na área de otimização numérica tanto de um ponto de vista teórico como aplicado, resultando no desenvolvimento de novos gradientes conjugados com melhores propriedades (Wang *et al.*, 2020; Kostopoulos e Grapsa, 2009; Zhang, 2009).

O método do gradiente conjugado de três termos é uma dessas variantes. Este método apresenta como um dos principais atrativos possuir um elevado ganho funcional por iteração, além de possuir convergência assegurada, mas com a desvantagem de necessitar de mais cômputos em cada iteração (Beale, 1972). A direção de busca para este método é dada por

$$\boldsymbol{z}_k = -\boldsymbol{g}_k + \delta_k \boldsymbol{z}_{k-1} + \mu_{k-1} \boldsymbol{z}_t, \tag{4.1}$$

em que, $\boldsymbol{g}_k$ representa o gradiente da função objetivo, δ_k indica o parâmetro conjugado, μ_{k-1} representa o parâmetro conjugado de três termos e $\boldsymbol{z}_t$ com $1 \leq t < k$ sendo a direção de reinício.

Tendo como referência o gradiente conjugado de três termos, propõe-se o algoritmo SMO conjugado de três termos LSSVM (*three-term conjugate-like LSSVM*, TCSMO-LSSVM) para a solução rápida do problema dual no treinamento de modelos LSSVM. Neste contexto, a

regra de atualização dos multiplicadores de Lagrange é dada por

$$\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^{k} + \rho_k z_k, \tag{4.2}$$

em que

$$z_k = \begin{cases} \boldsymbol{h}_{ij}^{k} + \delta_{k-1} z_{k-1} + \delta_{k-2} z_{k-2}, & se \;\; k \geq 1 \\ 0, \quad \text{demais casos.} \end{cases} \tag{4.3}$$

O vetor $\boldsymbol{h}_{ij}$ é determinado pela estratégia de seleção do conjunto de trabalho FGWSS, sendo definida como indicado abaixo

$$\boldsymbol{h}_{ij}^{k} = \boldsymbol{e}_i^k - \boldsymbol{e}_j^k = \begin{bmatrix} 0 \\ \vdots \\ 1_i \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 1_j \\ \vdots \\ 0 \end{bmatrix}. \tag{4.4}$$

Os termos 1_i e 1_j indicam valores unitários nas posições dos índices (i, j) selecionados pelo FGWSS. Avaliando as restrições de igualdade para os multiplicadores de Lagrange no problema dual, tem-se que

$$\sum_{i=1}^{n} \boldsymbol{\alpha}_i^{k+1} = \sum_{i=1}^{n} \boldsymbol{\alpha}_i^{k} + \rho_k [\sum_{i=1}^{n} \delta_{k-1} z_{k-1}^{i} + \sum_{i=1}^{n} \delta_{k-2} z_{k-2}^{i}] = 0, \tag{4.5}$$

portanto, para satisfazer a condição de igualdade do problema dual, deve-se ter

$$\sum_{i=1}^{n} z_{k-1}^{i} = 0 \tag{4.6}$$

e

$$\sum_{i=1}^{n} z_{k-2}^{i} = 0. \tag{4.7}$$

O parâmetro conjugado δ_{k-1} é consistente com a formulação obtida em Yu *et al.* (2023b) e é dado pela expressão

$$\delta_{k-1}^{*} = -\frac{\boldsymbol{h}_{ij}^{T} \tilde{\boldsymbol{K}} z_{k-1}}{z_{k-1}^{T} \tilde{\boldsymbol{K}} z_{k-1}}. \tag{4.8}$$

Considerando $\phi(\rho_k) = \mathscr{D}(\boldsymbol{\alpha}_k + \rho_k z_k)$ e lembrando que para a LSSVM, tem-se $\mathscr{D}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \tilde{K} \boldsymbol{\alpha} - \boldsymbol{y}^T \boldsymbol{\alpha}$, a partir da busca em linha exata (Watkins, 2004) é possível determinar o tamanho do passo ótimo em cada iteração da nova proposta, como indicado por

$$\begin{aligned}\phi'(\rho_k) &= z_k^T \nabla \mathscr{D}(\boldsymbol{\alpha}^k + \rho_k z_k) \\ &= z_k^T \boldsymbol{g}(\boldsymbol{\alpha}^k + \rho_k z_k) \\ &= z_k^T [\tilde{\boldsymbol{K}}(\boldsymbol{\alpha}^k + \rho_k z_k) - \boldsymbol{y}] \\ &= z_k^T \tilde{\boldsymbol{K}} \boldsymbol{\alpha}^k + \rho_k z_k^T \tilde{K} z_k - z_k^T \boldsymbol{y} \\ &= z_k^T (\tilde{\boldsymbol{K}} \boldsymbol{\alpha}^k - \boldsymbol{y}) + \rho_k z_k^T \tilde{\boldsymbol{K}} z_k \\ &= z_k^T \boldsymbol{g}(\boldsymbol{\alpha}^k) + \rho_k z_k^T \tilde{\boldsymbol{K}} z_k.\end{aligned} \tag{4.9}$$

Ao se considerar que $\phi'(\rho_k) = 0$, obtém-se o tamanho do passo ótimo como

$$\rho_k^* = -\frac{z_k^T \boldsymbol{g}(\alpha^k)}{z_k^T \tilde{\boldsymbol{K}} z_k}. \tag{4.10}$$

Com isso, substituindo a formulação para a direção de atualização z_k, mostrado na Equação (4.3), tem-se que o tamanho do passo ótimo pode ser dado por

$$\rho_k^* = -\frac{(\boldsymbol{h}_{ij}^k + \delta_{k-1} z_{k-1} + \delta_{k-2} z_{k-2})^T \boldsymbol{g}(\alpha^k)}{z_k^T \tilde{\boldsymbol{K}} z_k}, \tag{4.11}$$

ou seja,

$$\rho_k^* = \frac{-(\boldsymbol{h}_{ij}^k)^T \boldsymbol{g}(\alpha^k) - \delta_{k-1} z_{k-1}^T \boldsymbol{g}(\alpha^k) - \delta_{k-2} z_{k-2}^T \boldsymbol{g}(\alpha^k)}{z_k^T \tilde{\boldsymbol{K}} z_k}. \tag{4.12}$$

Sendo as direções z_{k-1} e z_{k-2} conjugadas, tem-se que $z_{k-1}^T \boldsymbol{g}(\alpha^k) = z_{k-2}^T \boldsymbol{g}(\alpha^k) = 0$, logo o tamanho do passo ótimo pode ser definido por

$$\rho_k^* = -\frac{\boldsymbol{g}^T(\alpha^k) \boldsymbol{h}_{ij}^k}{z_k^T \tilde{\boldsymbol{K}} z_k}. \tag{4.13}$$

Avaliando o denominador $z_k^T \tilde{K} z_k$, tem-se que a expressão resultante pode ser consideravelmente simplificada pelo uso da Equação (4.8) que fornece o parâmetro conjugado, δ_{k-1}^*, na atualização da direção de descida z_k como dado em

$$\begin{aligned}z_k^T \tilde{\boldsymbol{K}} z_k &= (\boldsymbol{h}_{ij} + \delta_{k-1} z_{k-1} + \delta_{k-2} z_{k-2})^T \tilde{\boldsymbol{K}} (\boldsymbol{h}_{ij} + \delta_{k-1} z_{k-1} + \delta_{k-2} z_{k-2}) \\ &= (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}^k + 2\delta_{k-1} (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} z_{k-1} + 2\delta_{k-2} (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} z_{k-2} + \\ &\qquad \delta_{k-1}^2 z_{k-1}^T \tilde{\boldsymbol{K}} z_{k-1} + \delta_{k-1}^2 z_{k-2}^T \tilde{\boldsymbol{K}} z_{k-2}.\end{aligned} \tag{4.14}$$

Substituindo a Equação (4.8) em (4.14), obtém-se a expressão quadrática em δ_{k-2} da seguinte forma

$$\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k = (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}^k - \frac{[(\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}]^2}{\boldsymbol{z}_{k-1}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}} + 2\delta_{k-2} (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2} + (\delta_{k-2})^2 \boldsymbol{z}_{k-2}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}. \tag{4.15}$$

Ao se considerar que $f_k(\delta_{k-2}) = \boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k$ é possível notar que a expressão resultante é quadrática e com fácil obtenção dos pontos extremos, uma vez que, coincidem com o vértice da parábola que representa a função $f_k(\delta_{k-2})$.

Note que o intuito de determinar δ_{k-2} que minimiza o termo $\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k$ é exatamente maximizar o ganho funcional por iteração, já que este termo é inversamente proporcional ao ganho para métodos que utilizam o algoritmo SMO com direções conjugadas como mostrado na Equação (3.18). A expressão quadrática dada pela Equação (4.15) pode ser simplificada pela introdução de termos auxiliares a_k, b_k e c_k, resultando na fórmula familiar abaixo

$$f_k(\delta_{k-2}) = a_k \delta_{k-2}^2 + 2b_k \delta_{k-2} + c_k, \tag{4.16}$$

em que,

$$\begin{cases} a_k = \boldsymbol{z}_{k-2}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}; \\ b_k = (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}; \\ c_k = (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}^k - \frac{[(\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}]^2}{\boldsymbol{z}_{k-1}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}}. \end{cases} \tag{4.17}$$

Uma vez que $a_k \geq 0$, $\forall k \geq 3$, existe um único valor mínimo de $f_k(\delta_{k-2})$ no ponto δ_{k-2}^*, dado pelas coordenadas

$$f_k(\delta_{k-2}^*) = c_k - \frac{b_k^2}{a_k} = (\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}^k - \frac{[(\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}]^2}{\boldsymbol{z}_{k-1}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}} - \frac{[(\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}]^2}{\boldsymbol{z}_{k-2}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}} \tag{4.18}$$

e

$$\delta_{k-2}^* = -\frac{b_k}{a_k} = -\frac{(\boldsymbol{h}_{ij}^k)^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}}{\boldsymbol{z}_{k-2}^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}}. \tag{4.19}$$

Teorema 4.1.1. *Se* $\boldsymbol{z}_q^T \tilde{\boldsymbol{K}} \boldsymbol{z}_q > 0 \quad (q = k-1, k)$, *então o ganho funcional do TCSMO é maior do que o ganho funcional do CSMO, como também do SMO padrão (Yu et al., 2023a).*

Demonstração. Com base na definição da direção $\boldsymbol{z}_k$ e estabelecendo que $\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k = \Delta_{TCSMO}^k$, então pela Equação (4.15), tem-se que

$$\Delta^k_{TCSMO} = (\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{h}^k_{ij} - \frac{[(\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}]^2}{\boldsymbol{z}^T_{k-1} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}} + 2\delta_{k-2} (\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2} + (\delta_{k-2})^2 \boldsymbol{z}^T_{k-2} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}. \tag{4.20}$$

Substituindo nessa expressão, δ^*_{k-2}, dado pela Equação (4.19), resulta na seguinte expressão

$$\Delta^k_{TCSMO} = (\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{h}^k_{ij} - \frac{[(\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}]^2}{\boldsymbol{z}^T_{k-1} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}} - \frac{[(\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}]^2}{\boldsymbol{z}^T_{k-2} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}} \leq (\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{h}^k_{ij} - \frac{[(\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}]^2}{\boldsymbol{z}^T_{k-1} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}} = \Delta^k_{CSMO}, \tag{4.21}$$

além disso, o ganho funcional do TCSMO é dado por

$$\mathscr{D}(\boldsymbol{\alpha}^k) - \mathscr{D}(\boldsymbol{\alpha}^{k+1}) = \frac{1}{2} \frac{[(\boldsymbol{h}^k_{ij})^T \boldsymbol{g}(\alpha^k)]^2}{\Delta^k_{TCSMO}} \geq \frac{1}{2} \frac{[(\boldsymbol{h}^k_{ij})^T \boldsymbol{g}(\alpha^k)]^2}{\Delta^k_{CSMO}} = f^{TCSMO}_G \geq f^{CSMO}_G, \tag{4.22}$$

logo, tem-se a seguinte relação de ordem $f^{TCSMO}_G \geq f^{CSMO}_G \geq f^{SMO}_G$. □

4.1.1 Implementação Computacional do TCSMO-LSSVM

Com intuito de reduzir o custo computacional, o TCSMO-LSSVM atualiza de forma incremental o gradiente e emprega o método do vetor auxiliar de uma forma similar ao que foi feito nos trabalhos do CSMO e TCSMO para o modelo SVM. De um forma geral, define-se a constante $\Psi^k = \boldsymbol{z}^T_k \tilde{\boldsymbol{K}} \boldsymbol{z}_k$ e o vetor auxiliar $\boldsymbol{v}^k = \tilde{\boldsymbol{K}} \boldsymbol{z}_k$. Com estas definições, os seguintes parâmetros são dados por expressões mais concisas e simplificadas, a saber (Torres-Barrán *et al.*, 2021; Yu *et al.*, 2023a)

$$\delta_{k-1} = -\frac{(\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}}{\boldsymbol{z}^T_{k-1} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-1}} = -\frac{(\boldsymbol{e}^k_i - \boldsymbol{e}^k_j)^T \boldsymbol{v}^{k-1}}{\Psi^{k-1}} = \frac{(\boldsymbol{v}^{k-1}_i - \boldsymbol{v}^{k-1}_j)}{\Psi^{k-1}}, \tag{4.23}$$

$$\delta_{k-2} = -\frac{(\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}}{\boldsymbol{z}^T_{k-2} \tilde{\boldsymbol{K}} \boldsymbol{z}_{k-2}} = -\frac{(\boldsymbol{e}^k_i - \boldsymbol{e}^k_j)^T \boldsymbol{v}^{k-2}}{\Psi^{k-2}} = \frac{(\boldsymbol{v}^{k-2}_i - \boldsymbol{v}^{k-2}_j)}{\Psi^{k-2}}, \tag{4.24}$$

$$\boldsymbol{v}^k = \tilde{\boldsymbol{K}} \boldsymbol{z}_k = \tilde{\boldsymbol{K}} (\boldsymbol{h}^k_{ij} + \delta_{k-1} \boldsymbol{z}_{k-1} + \delta_{k-2} \boldsymbol{z}_{k-2}) = [\tilde{\boldsymbol{K}}]_{:i} - [\tilde{\boldsymbol{K}}]_{:j} + \delta_{k-1} \boldsymbol{v}^{k-1} + \delta_{k-2} \boldsymbol{v}^{k-2}, \tag{4.25}$$

$$\Psi^k = \boldsymbol{z}^T_k \tilde{\boldsymbol{K}} \boldsymbol{z}_k = (\boldsymbol{h}^k_{ij} + \delta_{k-1} \boldsymbol{z}_{k-1} + \delta_{k-2} \boldsymbol{z}_{k-2})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k = (\boldsymbol{h}^k_{ij})^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k = \boldsymbol{v}^k_i - \boldsymbol{v}^k_j, \tag{4.26}$$

$$\rho_k = -\frac{\boldsymbol{g}^T(\boldsymbol{\alpha}^k \boldsymbol{h}_{ij}^k)}{\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k} = \frac{\boldsymbol{g}_j^k - \boldsymbol{g}_i^k}{\Psi^k}. \tag{4.27}$$

Do exposto, o detalhamento para implementação do TCSMO-LSSVM é apresentado pelo Algoritmo 7, enquanto o fluxograma da proposta é detalhado na Figura 5. Na linha 2, α^0, $\boldsymbol{z}_{-1}$, $\boldsymbol{z}_{-2}$, $\boldsymbol{v}_{-1}$ e $\boldsymbol{v}_{-2}$ são iniciados com o vetor nulo e com isso $\boldsymbol{g}(\boldsymbol{\alpha}_0) = \tilde{\boldsymbol{K}}\boldsymbol{\alpha}_0 - \boldsymbol{y} = -\boldsymbol{y}$, $\Psi^{-1} = \Psi^{-2} = 1$. No que se segue, a linha 4 indica o emprego da estratégia FGWSS para seleção do conjunto de trabalho, que é o par de índices (i, j) que determina quais multiplicadores de Lagrange, (α_i, α_j) devem passar por atualização em uma dada iteração.

Determinado o conjunto de trabalho para uma dada iteração, o parâmetro conjugado, δ_{k-1}, e o parâmetro conjugado de três termos, δ_{k-2}, são calculados utilizando a Equação (4.23) e a Equação (4.24), respectivamente. Neste ponto, percebe-se a importância do uso do vetor auxiliar $\boldsymbol{v}^k$ na simplificação tanto das equações como na implementação computacional.

Na linha 5 a direção conjugada para a k-ésima iteração pode ser construída com base nos parâmetros calculados na linha 4 e com as direções de dois passos anteriores, $\boldsymbol{z}_{k-1}$ e $\boldsymbol{z}_{k-2}$ utilizando a Equação (4.3). Uma vez que se tenha calculado a direção para a iteração correspondente, deve-se ter uma atualização das duas direções passadas, ou seja, $\boldsymbol{z}_{k-2} \Leftarrow \boldsymbol{z}_{k-1}$ e $\boldsymbol{z}_{k-1} \Leftarrow \boldsymbol{z}_k$.

Com todas as informações anteriores, é possível atualizar o vetor auxiliar utilizando a Equação (4.25) como mostrado na linha 6, com posterior atualização dos valores dos vetores auxiliares passados, ou seja, $\boldsymbol{v}^{k-2} \Leftarrow \boldsymbol{v}^{k-1}$ e $\boldsymbol{v}^{k-1} \Leftarrow \boldsymbol{v}^k$. Em seguida, a constante ψ^k é determinada pela Equação (4.26), com as posteriores atualizações $\psi^{k-2} \Leftarrow \psi^{k-1}$ e $\psi^{k-1} \Leftarrow \psi^k$.

Para finalizar a iteração, as linhas 8, 9, 10 e 11 correspondem ao cálculo do passo de atualização, ρ_k pela Equação (4.27), seguido pelas atualizações dos multiplicadores de Lagrange como dado pela Equação (4.2) e pela atualização do gradiente utilizando a expressão $\boldsymbol{g}(\boldsymbol{\alpha}^{k+1}) = \boldsymbol{g}(\boldsymbol{\alpha}^k) + \rho_k \boldsymbol{v}_k$.

Dado um determinado limiar ε, tem-se que estes procedimentos são repetidos até que a condição $max_i(g_i(\boldsymbol{\alpha})) - min_i(g_i(\boldsymbol{\alpha})) \leq \varepsilon$ seja alcançada ou até que o número máximo de iterações seja alcançado. Esta condição corresponde a $\|\boldsymbol{g}_k\| \leq \varepsilon$ ou que a norma do resíduo em uma dada iteração esteja abaixo do limiar ε.

Algoritmo 7: TCSMO-LSSVM

Entrada: $\boldsymbol{X}$, Limiar ε

início

Inicialize $\alpha^0 = \boldsymbol{z}_{-1} = \boldsymbol{z}_{-2} = \boldsymbol{v}^{-1} = \boldsymbol{v}^{-2} = \boldsymbol{0}$, $\boldsymbol{g}(\alpha^0) = -\boldsymbol{y}$, $\Psi^{-1} = \Psi^{-2} = 1$ e $k = 0$

while $max_i(g_i(\boldsymbol{\alpha})) - min_i(g_i(\boldsymbol{\alpha})) > \varepsilon$ **do**

Selecione o conjunto de trabalho (i, j) de acordo com 3.12; Calcule o parâmetro conjugado $\delta_{k-1} = \frac{(\boldsymbol{v}_i^{k-1} - \boldsymbol{v}_j^{k-1})}{\Psi^{k-1}}$ e o parâmetro conjugado de três termos $\delta_{k-2} = \frac{(\boldsymbol{v}_i^{k-2} - \boldsymbol{v}_j^{k-2})}{\Psi^{k-2}}$;

Construa a direção conjugada

$$z_k = \boldsymbol{h}_{ij}^k + \delta_{k-1} z_{k-1} + \delta_{k-2} z_{k-2}$$

$z_{k-2} \Leftarrow z_{k-1}$ e $z_{k-1} \Leftarrow z_k$;

Atualize o vetor auxiliar

$$\boldsymbol{v}^k = [\tilde{\boldsymbol{K}}]_{:i} - [\tilde{\boldsymbol{K}}]_{:j} + \delta_{k-1} v^{k-1} + \delta_{k-2} v^{k-2}$$

$\boldsymbol{v}^{k-2} \Leftarrow \boldsymbol{v}^{k-1}$ e $\boldsymbol{v}^{k-1} \Leftarrow \boldsymbol{v}^k$;

Calcule a constante

$$\Psi^k = \boldsymbol{v}_i^k - \boldsymbol{v}_j^k$$

$\boldsymbol{\Psi}^{k-2} \Leftarrow \boldsymbol{\Psi}^{k-1}$ e $\boldsymbol{\Psi}^{k-1} \Leftarrow \boldsymbol{\Psi}^k$;

Determine o tamanho do passo $\rho_k = \frac{\boldsymbol{g}_j^k - \boldsymbol{g}_i^k}{\Psi^k}$;

Atualize a variável dual $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k + \rho_k z_k$;

Atualize o gradiente $\boldsymbol{g}(\boldsymbol{\alpha}^{k+1}) = \boldsymbol{g}(\boldsymbol{\alpha}^k) + \rho_k \boldsymbol{v}_k$

Faça $k \Longleftarrow k+1$

end

fim

Resultado: Solução Dual $\boldsymbol{\alpha}^*$

4.1.2 Convergência

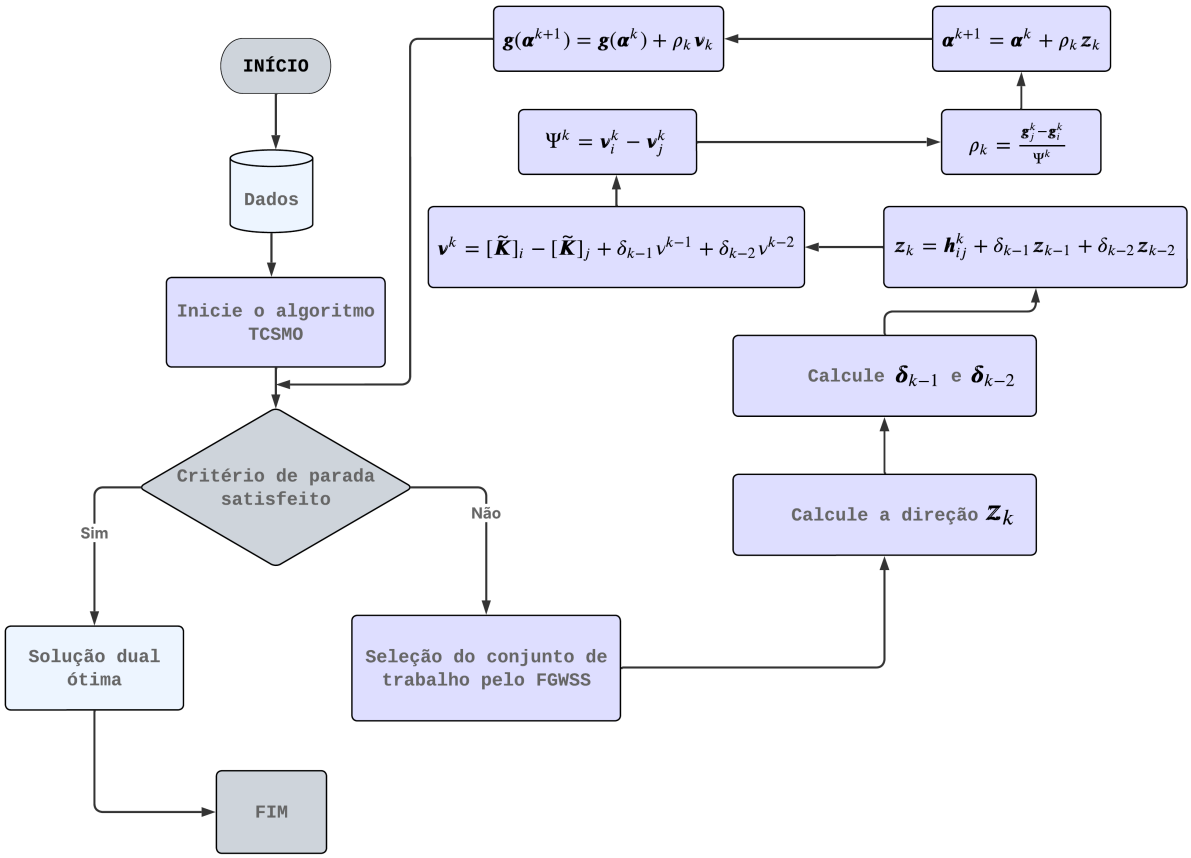
A demonstração da convergência do algoritmo segue de perto o mesmo racional adotado em Yu *et al.* (2023b). Para tal fim, é necessário estabelecer duas proposições.

Proposição 4.1. *Para o algoritmo TCSMO-LSSVM, o ganho funcional em duas iterações subsequentes satisfaz a seguinte desigualdade.*

$$f_G^{TCSMO}(\Delta\boldsymbol{\alpha}^k) \geq \frac{\left\|\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k+1}\right\|}{2\gamma} \tag{4.28}$$

Demonstração. Dado que o algoritmo SMO é convergente, a sequência, $\boldsymbol{\alpha}^k$, gerada pelas suas iterações é monótona decrescente e seu ganho funcional em duas iterações adjacentes satisfaz a

Figura 5 – Fluxograma para o algoritmo TCSMO.



Fonte: Elaborada pelo autor.

desigualdade

$$f_G^{SMO}(\Delta\boldsymbol{\alpha}^k) \geq \frac{\left\|\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k+1}\right\|}{2\gamma}. \tag{4.29}$$

Portanto, pelo Teorema 4.1.1, tem-se a desigualdade esperada dada por

$$f_G^{TCSMO} \geq f_G^{SMO} \geq \frac{\left\|\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k+1}\right\|}{2\gamma}. \tag{4.30}$$

□

Proposição 4.2. *A função dual $\mathscr{D}(\boldsymbol{\alpha})$ tem uma cota inferior $-2\frac{\sum_i y_i}{\sqrt{\lambda_{min}(\tilde{\boldsymbol{K}})}}$, em que $\lambda_{min}(\tilde{\boldsymbol{K}}) > 0$ indica o menor autovalor da matriz $\tilde{\boldsymbol{K}}$.*

Demonstração. De fato, a função dual satisfaz a desigualdade

$$\mathscr{D}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y}^T\boldsymbol{\alpha} \geq -\boldsymbol{y}^T\boldsymbol{\alpha}, \tag{4.31}$$

uma vez que pela desigualdade de Cauchy-Schwarz pode-se obter a seguinte relação

$$\boldsymbol{y}^T\boldsymbol{\alpha} \leq \|\,\boldsymbol{y}\|\,\|\boldsymbol{\alpha}\|\,, \tag{4.32}$$

e sabendo que $\lambda_{min}(\tilde{\boldsymbol{K}})\boldsymbol{\alpha}^T\boldsymbol{\alpha} \leq \boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha}$ então

$$\|\boldsymbol{\alpha}\| = \sqrt{\boldsymbol{\alpha}^T\boldsymbol{\alpha}} \leq \sqrt{\frac{\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha}}{\lambda_{min}(\tilde{\boldsymbol{K}})}}, \tag{4.33}$$

em razão disso

$$\boldsymbol{y}^T\boldsymbol{\alpha} \leq \|\boldsymbol{y}\|\,\|\boldsymbol{\alpha}\|\,| \leq \|\boldsymbol{y}\|\sqrt{\frac{\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha}}{\lambda_{min}(\tilde{\boldsymbol{K}})}}. \tag{4.34}$$

Note que $\mathbf{0}$ é solução factível do problema dual, o que implica que

$$\frac{1}{2}\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha} \leq \boldsymbol{\alpha}^T\boldsymbol{y} \leq \|\boldsymbol{y}\|\,|\sqrt{\frac{\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha}}{\lambda_{min}(\tilde{\boldsymbol{K}})}} \longrightarrow \sqrt{\boldsymbol{\alpha}^T\tilde{\boldsymbol{K}}\boldsymbol{\alpha}} \leq \frac{2\,\|\boldsymbol{y}\|}{\sqrt{\lambda_{min}(\tilde{\boldsymbol{K}})}}, \tag{4.35}$$

portanto, $\boldsymbol{\alpha}^T\boldsymbol{y} \leq \frac{2\|\boldsymbol{y}\|}{\sqrt{\lambda_{min}(\tilde{\boldsymbol{K}})}}$ e com isso

$$\mathscr{D}(\boldsymbol{\alpha}) \geq -\frac{2\,\|\boldsymbol{y}\|}{\sqrt{\lambda_{min}(\tilde{\boldsymbol{K}})}}. \tag{4.36}$$

□

Teorema 4.1.2. *A sequência $\{\boldsymbol{\alpha}_k\}$ gerada pelo algoritmo TCSMO-LSSVM converge para um ótimo global $\boldsymbol{\alpha}^*$ do problema dual.*

Demonstração. Pela Proposição 4.1, sabe-se que a função dual $\mathscr{D}(\boldsymbol{\alpha})$ é estritamente decrescente em todo o processo iterativo. Além disso, pela proposição 4.2, tem-se que $f_G^{TCSMO}(\Delta\boldsymbol{\alpha}^k)$ converge para 0, segue-se que a sequência $\{\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\}$ ainda converge para 0. Como $\tilde{\boldsymbol{K}}$ é uma matriz definida positiva simétrica, a função dual $\mathscr{D}(\boldsymbol{\alpha}^k)$ é estritamente convexa e com isso existe uma solução ótima global única $\boldsymbol{\alpha}^*$. Portanto, $\{\boldsymbol{\alpha}^k\}$ tem uma subsequência $\{\boldsymbol{\alpha}^{ks}\}$ com limite dado por $\hat{\boldsymbol{\alpha}}$. Sem perda de generalidade, suponha que o par de índices (i_F, j_F) foi selecionado, então considere o limite

$$\begin{aligned}\boldsymbol{g}_{iF}(\hat{\boldsymbol{\alpha}}) - \boldsymbol{g}_{jF}(\hat{\boldsymbol{\alpha}}) &= \lim_{ks\to\infty}(\boldsymbol{g}_{iF}(\boldsymbol{\alpha}^{ks}) - \boldsymbol{g}_{jF}(\boldsymbol{\alpha}^{ks})) \\ &= \lim_{ks\to\infty}(\boldsymbol{A}(ks) + \boldsymbol{B}(ks) + \boldsymbol{C}(ks)),\end{aligned} \tag{4.37}$$

em que

- $\boldsymbol{A}(ks) = \boldsymbol{g}_{iF}(\boldsymbol{\alpha}^{ks}) - \boldsymbol{g}_{iF}(\boldsymbol{\alpha}^{ks+1})$;
- $\boldsymbol{B}(ks) = \boldsymbol{g}_{iF}(\boldsymbol{\alpha}^{ks+1}) - \boldsymbol{g}_{jF}(\boldsymbol{\alpha}^{ks+1})$; e
- $\boldsymbol{C}(ks) = \boldsymbol{g}_{jF}(\boldsymbol{\alpha}^{ks+1}) - \boldsymbol{g}_{jF}(\boldsymbol{\alpha}^{ks})$.

Como $\{\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^{k}\}$ converge para 0, ambos $\boldsymbol{A}(ks)$ e $\boldsymbol{C}(ks)$ convergem para 0 quando $ks \longrightarrow \infty$. Além disso, $\boldsymbol{B}(ks) = 0$ é sempre satisfeito, então segue-se que $\boldsymbol{g}_{iF}(\hat{\boldsymbol{\alpha}}) - \boldsymbol{g}_{jF}(\hat{\boldsymbol{\alpha}}) = 0$. Considerando o método de seleção do conjunto de trabalho, pode-se tomar um outro limite e utilizando a proposição 4.1 chegar ao seguinte resultado

$$\begin{aligned}(\boldsymbol{g}_i(\hat{\boldsymbol{\alpha}}) - \boldsymbol{g}_j(\hat{\boldsymbol{\alpha}}))^2 &= (\lim_{ks\to\infty}(\boldsymbol{g}_i(\boldsymbol{\alpha}^{ks}) - \boldsymbol{g}_j(\boldsymbol{\alpha}^{ks})))^2, \quad \forall i, j \\ &\leq \frac{\boldsymbol{h}_{ij}^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}}{\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k} (\lim_{ks\to\infty}(\boldsymbol{g}_i(\boldsymbol{\alpha}^{ks}) - \boldsymbol{g}_j(\boldsymbol{\alpha}^{ks})))^2 \\ &= \frac{\boldsymbol{h}_{ij}^T \tilde{\boldsymbol{K}} \boldsymbol{h}_{ij}}{\boldsymbol{z}_k^T \tilde{\boldsymbol{K}} \boldsymbol{z}_k} (\boldsymbol{g}_{iF}(\hat{\boldsymbol{\alpha}}) - \boldsymbol{g}_{jF}(\hat{\boldsymbol{\alpha}}))^2 = 0, \quad \forall i, j.\end{aligned} \tag{4.38}$$

Com isso, tem-se que $\boldsymbol{g}_{iF}(\hat{\boldsymbol{\alpha}}) = \boldsymbol{g}_{jF}(\hat{\boldsymbol{\alpha}})$, $\forall i \neq j$, logo, tem-se que as condições KKT são satisfeitas e o valor limite $\hat{\boldsymbol{\alpha}}$ é a solução ótima global única de $\mathscr{D}(\boldsymbol{\alpha})$, já que a mesma é estritamente convexa.

□

4.1.3 Complexidade

Ao avaliar o algoritmo TCSMO-LSSVM em termos de complexidade, nota-se que o mesmo apresenta cálculos adicionais em cada iteração, quando comparado aos algoritmos SMO e CSMO. De fato, ao avaliar o SMO, observa-se uma complexidade de $2n$ operações de ponto flutuante por segundo (*floating-point operations per second*, flops) para a seleção do conjunto de trabalho e n flops para a atualização do gradiente. Para o algoritmo CSMO, a seleção do conjunto de trabalho também requer $2n$ flops para a sua realização e n flops para a atualização do gradiente.

Definindo n_v e n_z como o número de elementos não nulos do vetor auxiliar $\boldsymbol{v}$ e da direção conjugada $\boldsymbol{z}$, então a atualização de $\boldsymbol{z}$ e $\boldsymbol{\alpha}$ requerem $2n_z$ flops e a atualização do vetor auxiliar requer n_v flops. Destaca-se que $n_v \approx n$ e $n_z \ll n$. Portanto, para o CSMO são requeridas $2n_z + n_v + 3n \approx 4n$ flops.

O procedimento iterativo do TCSMO-LSSVM difere do CSMO apenas na atualização do vetor auxiliar $\boldsymbol{v}$ e direção $\boldsymbol{z}$ que usam informações de dois passos anteriores. Assim o

TCSMO-LSSVM necessita de $2n_v$ flops para atualizar $\boldsymbol{v}$ e $2n_z$ para atualizar $\boldsymbol{z}$, ou seja, para o processo completo requer $4n_z + 2n_v + 3n \approx 5n$.

Portanto, o algoritmo TCSMO-LSSVM é mais custoso por iteração do que o SMO e o CSMO. No entanto, o ganho funcional do TCSMO-LSSVM é superior aos dos demais como indicado pelo Teorema 4.1.1, ou seja, em poucas iterações a nova proposta alcança convergência. Um resumo dessa discussão é indicado na Tabela 3.

Tabela 3 – Custo em flops por iteração e ganho funcional de cada metodologia.

Algoritmo	**Flops**	**Ganho**
SMO	$3n$	Menor
CSMO	$4n$	Intermediário
TCSMO	$5n$	Maior

Fonte: elaborada pelo autor.

4.1.4 Poda Iterativa

O uso do algoritmo TCSMO para a solução do problema dual do LSSVM permite o rápido treinamento de modelos LSSVM, uma vez que tal algoritmo de otimização apresenta elevado ganho funcional por interação. No entanto, o problema relacionado a falta de esparsidade da solução ótima se mantém.

Para contornar tal desvantagem, empregou-se uma metodologia de poda que leva em consideração aqueles padrões que menos contribuíram para o ganho funcional durante o treinamento do LSSVM. O racional empregado é similar ao apresentado em Zeng e Chen (2005). Para fins de entendimento, considere o problema dual do LSSVM como dado pela Equação (2.26) e a condição KKT dada pela expressão

$$\nabla \mathscr{D}(\boldsymbol{\alpha}) = \boldsymbol{g}(\boldsymbol{\alpha}) = \tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y} = 0 \longrightarrow F_i = \sum_{j=1}^{n} \alpha_j \tilde{K}_{ij} - y_i \tag{4.39}$$

Para derivar um critério para a poda, a função objetivo dual é reescrita usando a definição de F_i, como dado abaixo

$$\mathscr{D}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_i \alpha_i (y_i - F_i). \tag{4.40}$$

Se a amostra k é removida do conjunto de treino, a função objetivo deve passar por

atualização para considerar somente as amostras restantes, ou seja

$$\mathscr{D}(\boldsymbol{\alpha'}) = \frac{1}{2}\sum_{i\neq k}\alpha_i(y_i - F_i'), \tag{4.41}$$

em que a k-ésima amostra não é computada no somatório para a obtenção de $\mathscr{D}(\boldsymbol{\alpha'})$, $\boldsymbol{\alpha'} = (\alpha_1,\ldots,\alpha_{k-1},\alpha_{k+1},\ldots,\alpha_n)$ e $F_i' = F_i - \alpha_k\tilde{K}_{ik}$. Considerando que a remoção de uma amostra k não afeta diretamente os vetores de suporte das outras amostras, mas introduz a atualização de todo F_i, conduzindo assim a diferença $d(\mathscr{L})$ na função objetivo, derivada pela expressão

$$\begin{aligned}
d(\mathscr{D}) &= \frac{1}{2}\sum_{i}\alpha_i(y_i - F_i) - \frac{1}{2}\sum_{i\neq k}\alpha_i(y_i - F_i') \\
&= \frac{1}{2}\sum_{i\neq k}\alpha_i(y_i - F_i) + \frac{1}{2}\alpha_k(y_k - F_k) - \frac{1}{2}\sum_{i\neq k}\alpha_i(y_i - F_i') \\
&= \frac{1}{2}\sum_{i\neq k}\alpha_i(F_i' - F_i) + \frac{1}{2}\alpha_k(y_k - F_k) \\
&= \frac{-1}{2}(\sum_{i}\alpha_i\alpha_k\tilde{K}_{ik} - \alpha_k^2\tilde{K}_{kk}) + \frac{1}{2}\alpha_k(y_k - F_k) \\
&= \frac{1}{2}\alpha_k^2\tilde{K}_{kk} - \alpha_k F_k.
\end{aligned} \tag{4.42}$$

Com o intuito de considerar apenas os padrões que mais contribuem para a minimização da função custo dual, $\mathscr{D}(.)$, retira-se o padrão k que apresenta o menor valor de $d(\mathscr{D})$. Além disso, é importante destacar que a poda de uma amostra com posterior retreinamento não é eficiente em contextos de processamento de grandes bases de dados. Portanto, o procedimento de poda é realizado como um pós-processamento ao treino, permitindo a obtenção de uma solução esparsa sem grandes impactos na performance preditiva do modelo. No Algoritmo 8 são detalhados os principais pontos relacionados ao processo de poda iterativa do modelo proposto.

As linhas 2 à 7 consistem na inicialização do algoritmo. Nas linhas 1 e 2 são construídos os conjuntos ativo e inativo. Inicialmente, o conjunto ativo é formado pelo conjunto $I_N = \{1,2,\ldots,N\}$, em que N representa o número de padrões de treinamento, já o conjunto inativo é iniciado sem elementos, em outras palavras, o conjunto ativo é iniciado com todos os padrões de treino e o inativo é vazio.

A linha 5 mostra a expressão para o cálculo do termo de erro $E = \max(\boldsymbol{g}(\boldsymbol{\alpha})) - \min(\boldsymbol{g}(\boldsymbol{\alpha}))$ que funciona como um indicativo do valor da norma do gradiente, ou ainda, a norma do resíduo. Vale frisar que este termo é calculado com base na estimativa final do vetor gradiente ao fim do treinamento do TCSMO-LSSVM.

Na linha 7 o termo $\boldsymbol{F}$ é iniciado utilizando a Equação (4.39). Em seguida, o laço principal do algoritmo é iniciado. Na linha 9 o termo de diferença, $d(\mathscr{D})$, é calculado para cada padrão de treinamento utilizando a Equação (4.42), sendo as N_r amostras com menores valores de $d(\mathscr{D})$ podadas.

Uma vez que a poda tenha ocorrido, deve-se ter uma atualização dos conjuntos ativo e inativo. Os índices dos padrões podados são transferidos do conjunto ativo para o inativo. A linha 12 apresenta a expressão para a atualização do valor de F, uma vez que padrões foram retirados o valor de F deve ser recalculado para considerar apenas o conjunto ativo atualizado.

A atualização do gradiente deve ser realizada sempre que ocorre atualizações nos conjuntos ativo e inativo. O procedimento completo se repete até que o número máximo de iterações tenha sido alcançado ou caso os valores do termo de erro em iterações sucessivas fique abaixo de um dado limiar.

Algoritmo 8: Poda baseada no ganho funcional

Entrada: Solução não esparsa $\boldsymbol{\alpha}^*$, gradiente $\boldsymbol{g}(\boldsymbol{\alpha})$, Número de padrões podados por iteração, N_r, Número máximo de iterações K e Limiar ε

1 **início**

2 Inicie o conjunto ativo com os índices de 1 ao número de padrões de treino;

3 Inicie o conjunto inativo como um conjunto vazio;

4 Inicie o erro, E usando a expressão.

5 $$E = \max(\boldsymbol{g}(\boldsymbol{\alpha})) - \min(\boldsymbol{g}(\boldsymbol{\alpha}))$$

6 Calcule o vetor $\boldsymbol{F}$;

7 $$\boldsymbol{F} = \tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y}$$

8 **while** $k <= K \quad \& \quad E_k - E_{k-1} <= \varepsilon$ **do**

9 Calcule a diferença $d(\mathscr{L})$ e remova os N_r padrões com menores valores para $d(\mathscr{L})$;

10 Atualize os conjuntos ativo e inativo;

11 Atualize o vetor F;

12 $$F = F - \tilde{\boldsymbol{K}}\alpha[inativo, :]$$

Atualize o gradiente;

13 $$\boldsymbol{g}(\boldsymbol{\alpha}) = \tilde{\boldsymbol{K}}\alpha[ativo, :] - y$$

Faça $k \Longleftarrow k+1$

14 **end**

15 **fim**

Resultado: Solução esparsa $\boldsymbol{\alpha}^*$

4.2 Gradiente Conjugado Espectral LSSVM (Proposta 2)

O método dos gradientes conjugados trouxe vários benefícios quando comparado ao seu antecessor gradiente descendente, como ortogonalidade dos resíduos, conjugação das direções, direções anteriores não precisam ser recalculadas e custo somente um pouco superior ao gradiente descendente por iteração. No entanto, o emprego de busca em linha exata no cálculo do passo de atualização ótimo requer a realização de produtos vetor-matriz, que em cenários de alta volumetria pode requerer alto custo de processamento (Barzilai e Borwein, 1988; Birgin e Martínez, 2001). Visando alternativas à busca em linha exata, os métodos espectrais são propostos com a ideia de utilizar uma informação de segunda ordem aproximada na estimativa do passo ótimo.

4.2.1 Métodos Espectrais

Uma alternativa eficiente às buscas lineares tradicionais foi proposto em Barzilai e Borwein (1988). O método desenvolvido utiliza informações de duas iterações consecutivas para calcular o passo, α, baseado em uma aproximação da curvatura local da função, quase sem custos adicionais. De fato, a metodologia apresenta comportamento similar aos métodos de segunda ordem (Newton ou quase-Newton), mas sem calcular e nem armazenar a Hessiana.

Dado os valores de duas iterações consecutivas de uma variável de otimização $\boldsymbol{x}$, ou seja, conhecendo os valores $\boldsymbol{x}_k$ e $\boldsymbol{x}_{k-1}$, defina o deslocamento no espaço das variáveis como $\boldsymbol{s}_{k-1} = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ e a variação no gradiente da função objetivo, $\boldsymbol{f}$, como $\boldsymbol{y}_k = \nabla \boldsymbol{f}(\boldsymbol{x}_k) - \nabla \boldsymbol{f}(\boldsymbol{x}_{k-1})$, então pelo método de Barzilai–Borwein, o passo pode ser dado pelas expressões que se seguem

$$\alpha_k^1 = \frac{\boldsymbol{s}_{k-1}^T \boldsymbol{s}_{k-1}}{\boldsymbol{s}_{k-1}^T \boldsymbol{y}_{k-1}} \tag{4.43}$$

e

$$\alpha_k^2 = \frac{\boldsymbol{s}_{k-1}^T \boldsymbol{y}_{k-1}}{\boldsymbol{y}_{k-1}^T \boldsymbol{y}_{k-1}}. \tag{4.44}$$

Os valores α_k^1 e α_k^2 são ainda conhecidos como passos BB_1 e BB_2, respectivamente. Em linhas gerais, tem-se que:

- O uso de BB_1 favorece avanços rápidos, mas menos estáveis;
- O BB_2 favorece avanços mais graduais e mais estáveis.

Uma das propriedades dos métodos espectrais que o fazem ser similar aos métodos quase-Newton está no fato de utilizarem a condição de secante para métodos quase-Newton

(Broyden *et al.*, 1973; Broyden, 1973). Esta condição afirma que a matriz que aproxima a Hessiana, $\boldsymbol{B}_k$, ou sua inversa, $\boldsymbol{B}_k^{-1}$, devem satisfazer a relação

$$\boldsymbol{B}_{k+1}\boldsymbol{s}_k = \boldsymbol{y}_k, \tag{4.45}$$

em que essa condição é uma generalização da aproximação de primeira derivada usada no método da secante unidimensional e garante que a nova aproximação da matriz (ou sua inversa) "lembre"a informação do último passo de iteração, aproximando-se da verdadeira Hessiana (Broyden, 1973). No caso de métodos espectrais, a ideia é utilizar o escalar $\frac{1}{\alpha_k}$ para aproximar a Hessiana $\boldsymbol{B}_k$, ou ainda

$$\frac{1}{\alpha_{k+1}}\boldsymbol{s}_k = \boldsymbol{y}_k, \tag{4.46}$$

em que o passo α_k é selecionado de forma a minimizar o erro quadrático dessa aproximação, levando à Equação (4.43) e Equação (4.44). A aproximação dada pela Equação (4.46) justifica o termo 'espectral' na denominação dos métodos. O termo $\frac{1}{\alpha_k}$ atua como uma estimativa espectral do autovalor dominante da Hessiana na direção atual o que torna o método mais sensível à geometria da superfície da função objetivo, justificando a principal desvantagem destes métodos que é a falta de estabilidade, fazendo que este oscile de forma considerável em torno de uma solução.

4.2.2 Métodos dos Gradientes Conjugados Espectrais

Os métodos dos gradientes conjugados espectrais traz os benefícios de redução do custo computacional na estimativa do passo de atualização, uma vez que, utiliza as informações locais da curvatura da função objetivo, sem o uso de busca linear ao mesmo tempo que utiliza direções construídas por conjugação como no caso do método dos gradientes conjugados padrão, favorecendo a robustez e estabilidade durante o processo de otimização (Birgin e Martínez, 2001).

O diferencial destes métodos quando comparado aos métodos dos gradientes conjugados padrão, está na estimativa do passo de atualização que é dado pela uso da Equação (4.43), ao invés do uso de busca em linha exata como dado na Equação (2.36). No que se segue a metodologia é similar ao detalhado na subseção 2.5.2. Com esta adaptação, o método resultante é caracterizado por:

– Convergência mais rápida do que métodos CG padrões;

– Robustez em otimização de funções mal condicionadas;
– Mantém a estrutura conjugada das direções, garantindo eficiência em problemas quadráticos.

Estes métodos são muitas vezes considerados precursores de otimizadores adaptativos modernos e frequentemente empregados em tarefas de ajuste de redes neurais profundas, como os otimizadores de subgradientes adaptativos (*adaptive subgradient*, AdaGrad) (Duchi *et al.*, 2011), de propagação da raiz quadrática média (*root mean square propagation*, RMSProp) (Kurbiel e Khaleghian, 2017) e o de estimação de momentos adaptativos (*adaptive moment estimation*, Adam) (Zhang, 2018). No entanto, o problema da estabilidade associado ao uso do passo de atualização aproximado pela curvatura local da função ainda pode ser considerável e impacta negativamente no processo de otimização fazendo com que soluções subótimas sejam obtidas.

4.2.3 Ajuste de LSSVMs via Gradientes Conjugados Espectrais

A segunda proposta denominada gradiente conjugado espectral LSSVM (*spectral conjugate gradient LSSVM*, SCG-LSSVM), consiste no uso de um recente método do gradiente conjugado espectral para a solução do problema dual no treinamento de LSSVMs.

O novo esquema de escolha dos parâmetros espectral e conjugado favorece o desenvolvimento de uma nova direção de busca que satisfaz a propriedade espectral e a condição de descida simultaneamente, favorecendo alto ganho funcional a cada iteração. Além disso, neste novo algoritmo, emprega-se o uso de informação de segunda ordem por meio da aproximação de Broyden-Fletcher-Goldfarb-Shanno para a Hessiana da função objetivo dual o que contribui para uma rápida convergência sem grande custos de processamento por iteração.

A aproximação BFGS para a Hessiana é dada em termos do deslocamento no espaço de variáveis e na variação do gradiente da função objetivo dual, além disso, o passo de atualização é dado pelo uso do método de busca em linha de Wolfe forte (Pérez e Prudente, 2019) o que significa que a condição de Armijo forte, que consiste em selecionar um passo ρ de atualização que satisfaça a relação que se segue deve ser satisfeita

$$\boldsymbol{f}(\boldsymbol{\alpha}_k + \rho_k \boldsymbol{d}_k) \leq \boldsymbol{f}(\alpha_k) + c_1 \rho_k \boldsymbol{g}^T(\boldsymbol{\alpha}_k)\boldsymbol{d}_k. \tag{4.47}$$

Além da condição de Armijo, na busca em linha de Wolfe forte, o termo de passo também deve

satisfazer a condição de curvatura, ou seja

$$|\boldsymbol{g}_k^T(\boldsymbol{\alpha}_k+\rho_k\boldsymbol{d}_k)\boldsymbol{d}_k| \leq c_2|\boldsymbol{g}_k^T(\boldsymbol{\alpha}_k)\boldsymbol{d}_k|, \tag{4.48}$$

ou de forma resumida,

$$|\boldsymbol{g}_{k+1}^T\boldsymbol{d}_k| \leq c_2|\boldsymbol{g}_k^T\boldsymbol{d}_k|, \tag{4.49}$$

em que, $0 < c_1 \leq c_2 < 1$.

Diante do exposto, considera-se a equação iterativa dada pela Equação (4.2). Uma vez que, a direção de descida combina o método do gradiente conjugado e o método do gradiente espectral, a mesma pode ser dada pela combinação linear a seguir (Hmede *et al.*, 2022b)

$$\boldsymbol{d}_k = -\theta_k\boldsymbol{g}_k + \beta_k\boldsymbol{s}_{k-1}, \tag{4.50}$$

em que $\boldsymbol{s}_k = \boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k-1}$, $\boldsymbol{g}_k = \nabla f(\boldsymbol{\alpha}_k)$ e, finalmente, θ_k e β_k são os parâmetros espectral e conjugado, respectivamente. Seja $\overline{\boldsymbol{d}}_k$ a direção de descida do método do gradiente conjugado clássico, além disso, considere a aproximação de Taylor de 2° ordem para a função objetivo, como dado por

$$\psi(\rho) = f(\boldsymbol{\alpha}_k+\rho\overline{\boldsymbol{d}}_k) = f(\boldsymbol{\alpha}_k)+\rho\boldsymbol{g}_k^T\overline{\boldsymbol{d}}_k+\frac{1}{2}\rho^2\overline{\boldsymbol{d}}_k^T\boldsymbol{B}_k\overline{\boldsymbol{d}}_k, \tag{4.51}$$

em que $\boldsymbol{B}_k$ representa a matriz Hessiana da função objetivo no ponto $\boldsymbol{\alpha}_k$. Com o intuito de encontrar o passo, ρ_k^*, que minimiza a função objetivo na direção de descida correspondente, faz-se $\psi'(\rho) = 0$. Desta forma o valor ótimo para o passo na k-ésima iteração, é dado por

$$\rho_k^* = \frac{-\boldsymbol{g}_k^T\overline{\boldsymbol{d}}_k}{\overline{\boldsymbol{d}}_k^T\boldsymbol{B}_k\overline{\boldsymbol{d}}_k}. \tag{4.52}$$

Dado que para o cálculo da matriz Hessiana $\boldsymbol{B}_k$ é necessário a determinação de derivadas de 2° ordem, uma estratégia mais eficiente para sua estimação é necessária. Seguindo o mesmo procedimento do trabalho de Wang *et al.* (2020), utilizou-se a regra de atualização BFGS para o cálculo iterativo da Hessiana (Babaie-Kafaki, 2015; Nocedal, 1980)

$$\boldsymbol{B}_k = \boldsymbol{B}_{k-1} - \frac{\boldsymbol{B}_{k-1}\boldsymbol{s}_{k-1}\boldsymbol{s}_{k-1}^T\boldsymbol{B}_{k-1}}{\boldsymbol{s}_{k-1}^T\boldsymbol{B}_{k-1}\boldsymbol{s}_{k-1}} + \frac{\boldsymbol{l}_{k-1}\boldsymbol{l}_{k-1}^T}{\boldsymbol{s}_{k-1}^T\boldsymbol{l}_{k-1}}, \tag{4.53}$$

em que $\boldsymbol{l}_{k-1} = \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$. Com o intuito de reduzir os custos computacionais e de armazenamento, os esquemas BFGS sem memória (*memoryless*) são geralmente usados para substituir $\boldsymbol{B}_k$ (Wang *et al.*, 2020). Neste trabalho, foi utilizado $\boldsymbol{B}_{k-1}$ como uma matriz escalar dada por

$$\varphi\frac{||\boldsymbol{l}_{k-1}||^2}{\boldsymbol{s}_{k-1}^T\boldsymbol{l}_{k-1}}\boldsymbol{I}. \tag{4.54}$$

Então substituindo a Equação (4.54) na Equação (4.53), resulta na expressão abaixo, que representa a regra de atualização final para a matriz Hessiana a cada iteração.

$$\boldsymbol{B}_k = \varphi \frac{||\boldsymbol{l}_{k-1}||^2}{\boldsymbol{s}_{k-1}^T \boldsymbol{l}_{k-1}} \boldsymbol{I} - \varphi \frac{||\boldsymbol{l}_{k-1}||^2}{\boldsymbol{s}_{k-1}^T \boldsymbol{l}_{k-1}} \frac{\boldsymbol{s}_{k-1}\boldsymbol{s}_{k-1}^T}{\boldsymbol{s}_{k-1}^T \boldsymbol{s}_{k-1}} + \frac{\boldsymbol{l}_{k-1}\boldsymbol{l}_{k-1}^T}{\boldsymbol{s}_{k-1}^T \boldsymbol{l}_{k-1}} \tag{4.55}$$

Para o caso de $\boldsymbol{s}_{k-1}^T \boldsymbol{l}_{k-1} > 0$, tem-se que $\boldsymbol{B}_k$ é simétrica e definida positiva. Se a direção $\overline{\boldsymbol{d}}_k$ é selecionada pela Fórmula DY (Dai e Yuan, 1999), i.e.:

$$\overline{\boldsymbol{d}}_k = \boldsymbol{d}_k^{DY} = -\boldsymbol{g}_k + \beta_k^{DY} \boldsymbol{s}_{k-1}, \tag{4.56}$$

com $\beta_k^{DY} = \frac{||\boldsymbol{g}_k||^2}{\boldsymbol{s}_{k-1}^T \boldsymbol{l}_{k-1}}$. Substituindo as equações (4.56) e (4.55) na Equação (4.52), obtem-se a expressão final para o tamanho do passo ótimo aproximado

$$\rho_k^* = \frac{-\boldsymbol{s}_{k-1}^T \boldsymbol{g}_{k-1}}{\varphi ||\boldsymbol{l}_{k-1}||^2 \boldsymbol{p}_k}, \tag{4.57}$$

em que,

$$\boldsymbol{p}_k = 1 - \frac{(\boldsymbol{g}_k^T \boldsymbol{s}_{k-1})^2}{||\boldsymbol{g}_k||^2 ||\boldsymbol{s}_{k-1}||^2} + \frac{1}{\varphi} \left(\frac{\boldsymbol{g}_k^T \boldsymbol{l}_{k-1}}{||\boldsymbol{g}_k|| ||\boldsymbol{l}_{k-1}||} + \frac{||\boldsymbol{g}_k||}{||\boldsymbol{l}_{k-1}||} \right)^2. \tag{4.58}$$

Para garantir que a propriedade de descida suficiente é satisfeita, bem como, a propriedade de limitação do parâmetro espectral θ_k, a técnica de truncamento desenvolvida em Liu e Liu (2018) é adotada para selecionar θ_k e β_k, como dado por

$$\theta_k = \max\{\min\{\rho_k^*, \overline{\phi}_k\}, \phi_k\} \tag{4.59}$$

e

$$\beta_k = \theta_k \beta_k^{DY}, \tag{4.60}$$

em que $\overline{\phi}_k = \frac{||\boldsymbol{s}_{k-1}||^2}{\boldsymbol{s}_{k-1}^T \boldsymbol{l}_{k-1}}$ e $\phi_k = \frac{\boldsymbol{s}_{k-1}^T \boldsymbol{l}_{k-1}}{||\boldsymbol{l}_{k-1}||^2}$. O Algoritmo 9 resume todo o detalhamento matemático anterior de uma forma sucinta e adequada para implementação numérica. A Figura 6 apresenta um fluxograma destacando a sequência de etapas para a execução do SCG-LSSVM.

Uma vez que são dados os valores do limiar de precisão ε para a solução e o valor da aproximação inicial da solução $\boldsymbol{\alpha}_0$, as linhas 2 e 3 do algoritmo consistem apenas no cálculo dos valores iniciais para a função objetivo e para o gradiente. Com estas informações, é possível calcular o valor do passo ótimo com base na Equação (4.57), analisando se as duas condições de

Wolfe são satisfeitas, do contrário inicie com $\rho = 1$ e teste a condição de Armijo, caso esta não seja satisfeita reduza o valor do passo até que a condição seja satisfeita. Em seguida, verifique a condição da curvatura, caso esta não seja satisfeita, amplie um pouco o valor do passo até que ambas as condições sejam satisfeitas.

Na linha 6 os multiplicadores de Lagrange são atualizados utilizando a Equação (4.2). Na linha 7 os parâmetros espectral e conjugado são determinado via as equações (4.59) e (4.60) e, por fim, a direção de descida é atualizada utilizando a Equação (4.50). O processo completo é repetido até que a norma do gradiente da função objetivo esteja abaixo do valor limiar ε.

Algoritmo 9: SCG-LSSVM

Entrada: $\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}_0, \varepsilon$ e $1 \leq \phi \leq 2$

1 **início**
2 Inicie $k = 0$;
3 Calcule $f(\boldsymbol{\alpha}_0) = f_0 = \frac{1}{2}\boldsymbol{\alpha}_0^T \tilde{\boldsymbol{K}} \boldsymbol{\alpha}_0 - \boldsymbol{y}^T \boldsymbol{\alpha}_0$ e $\boldsymbol{g}_0 = \nabla f(\boldsymbol{\alpha}_0) = \tilde{\boldsymbol{K}} \boldsymbol{\alpha}_0 - \boldsymbol{y}$;
4 **while** $||\boldsymbol{g}_k|| \leq \varepsilon$ **do**
5 Calcule o tamanho do passo $\boldsymbol{\rho}_k$ usando a busca linear como dado na Equação (4.57);
6 Atualize a variável dual, fazendo $\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \rho_k^* \boldsymbol{d}_k$ e o gradiente $\boldsymbol{g}_{k+1}$;
7 Calcule os parâmetros θ_k e β_k usando a fórmula de truncamento apresentada na Equação (4.59) e pela Equação (4.60);
8 Atualize a direção de descida $\boldsymbol{d}_{k+1}$ utilizando a Equação (4.50);
9 faça $k \longleftarrow k+1$.
10 Retorne ao PASSO 5.
11 **end**
12 **fim**

Resultado: Multiplicadores de Lagrange ótimos $\boldsymbol{\alpha}^*$

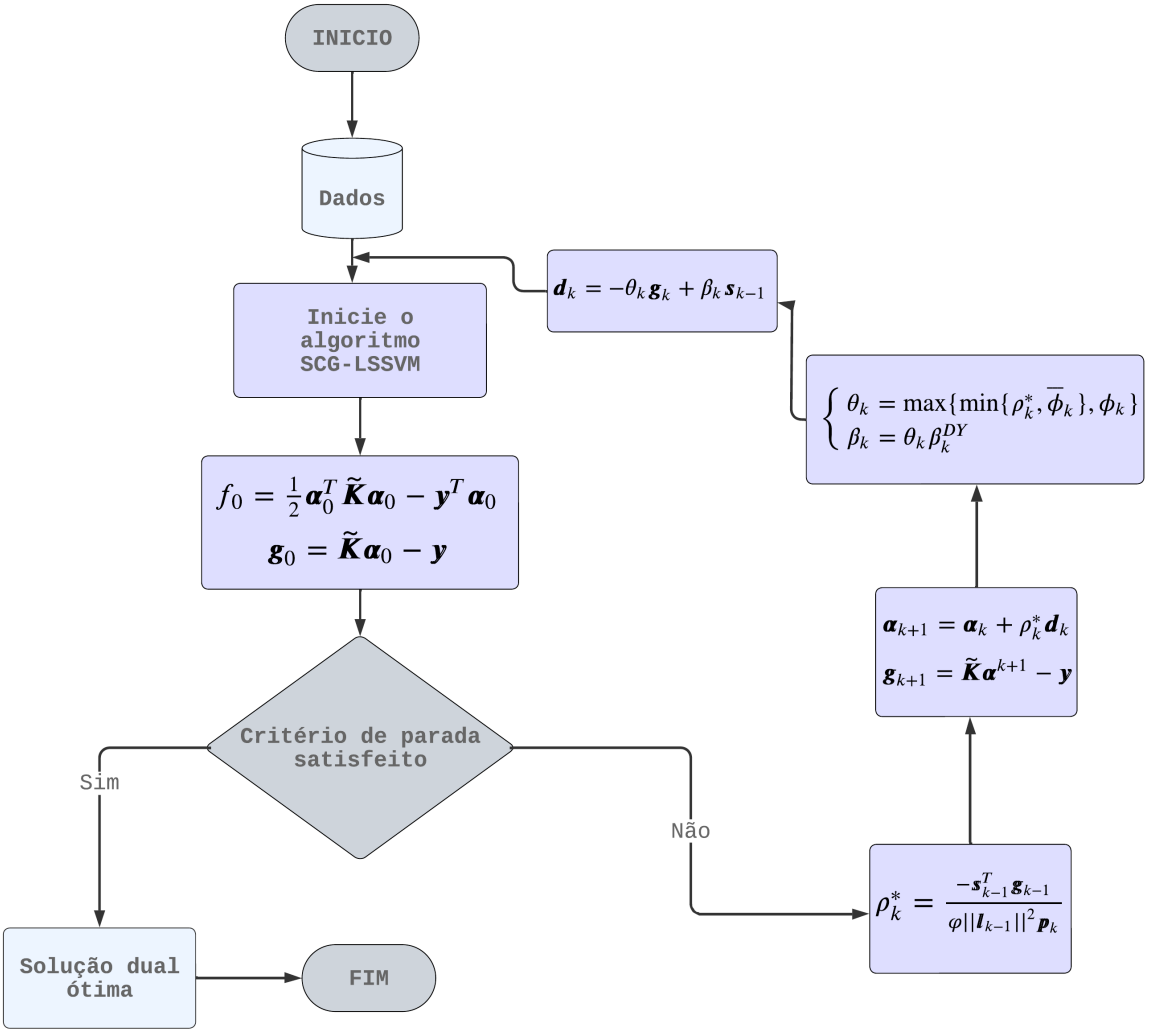
4.2.4 Convergência

Seguindo a fundamentação estabelecida em Wang *et al.* (2020), o primeiro passo para analisar a convergência do modelo SCG-LSSVM é provar que a direção dada pela Equação (4.50) é uma direção de descida, conforme enunciado no teorema abaixo.

Teorema 4.2.1. *A direção de busca dada pela Equação (4.50) é uma direção de descida suficiente, i.e,*

$$\boldsymbol{g}_k^T \boldsymbol{d}_k \leq -K||\boldsymbol{g}_k||^2, \tag{4.61}$$

Figura 6 – Fluxograma para o algoritmo SCG.



Fonte: Elaborada pelo autor.

Demonstração. Com $K \in \mathbb{R}$, além disso, como o parâmetro θ_k é dado pela regra de truncamento indicada pela Equação (4.59), então θ é limitado, ou seja, $\exists \;\; M, m \in \mathbb{R}^+$, tal que

$$m \leq \theta_k \leq M. \tag{4.62}$$

Pelo fato de a busca do tamanho do passo ser feita por meio de busca em linha de Wolfe, tem-se que as iterações do gradiente satisfazem a seguinte expressão (Pérez e Prudente, 2019).

$$|\boldsymbol{g}_{k+1}^T \boldsymbol{d}_k| \leq c|\boldsymbol{g}_k^T \boldsymbol{d}_k| \longrightarrow t_k = \frac{\boldsymbol{g}_k^T \boldsymbol{s}_{k-1}}{\boldsymbol{g}_{k+1}^T \boldsymbol{s}_{k-1}} \;\; \in [-c, c]. \tag{4.63}$$

Além disso, pré-multiplicando a Equação (4.50) por $\boldsymbol{g}_k^T$, tem-se que

$$\begin{aligned} \boldsymbol{g}_k^T \boldsymbol{d}_k &= -\theta_k ||\boldsymbol{g}_k||^2 + \beta_k \boldsymbol{g}_k^T \boldsymbol{s}_{k-1} \\ = \theta_k ||\boldsymbol{g}_k||^2 \frac{1}{t_k - 1} &\leq -\frac{m}{1+c} ||\boldsymbol{g}_k||^2 = -K||\boldsymbol{g}_k||^2, \end{aligned} \tag{4.64}$$

em que $K = \frac{m}{1+c}$.

□

Além disso, é importante considerar algumas características próprias do problema dual para o treinamento da LSSVM, delineadas abaixo:

- O conjunto de nível $\mathscr{N} = \{\boldsymbol{\alpha} / \mathscr{D}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y}^T\boldsymbol{\alpha} \leq \boldsymbol{\alpha}^T \tilde{\boldsymbol{K}}\boldsymbol{\alpha}\}$ é limitado;
- A função objetivo $\mathscr{D}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{y}^T\boldsymbol{\alpha}$ é continuamente diferenciável e seu gradiente é *lipschitz*, i.e., existe uma constante $L > 0$, tal que $||\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{y})|| \leq L||\boldsymbol{x} - \boldsymbol{y}||$.

Estas características do problema dual, permitem inferir que existe uma constante $\Gamma \geq 0$ tal que

$$||\boldsymbol{g}(\boldsymbol{x})|| \leq \Gamma, \quad para \quad um \quad x \in \mathscr{N}. \tag{4.65}$$

A seguinte proposição foi originalmente dada em Wang *et al.* (2020) e será útil para a demonstração da convergência do algoritmo SCG.

Proposição 4.3. *As sequências* $\{\boldsymbol{d}_k\}$ *e* $\{\rho_k\}$ *geradas pelo algoritmo SCG satisfazem.*

$$\sum_{k=0}^{\infty} \frac{(\boldsymbol{g}_k^T \boldsymbol{d}_k)^2}{||\boldsymbol{d}_k||^2} < \infty. \tag{4.66}$$

A proposição 4.3 também é conhecida como condição de Zoutendijk e foi originalmente proposta em Zoutendijk (1970), onde o detalhamento matemático de sua prova pode ser encontrado.

Proposição 4.4. *Sejam as sequências* $\{\boldsymbol{d}_k\}$ *e* $\{\rho_k\}$ *geradas pelo algoritmo SCG, então são satisfeitas as seguintes convergências*

$$\liminf_{k \longrightarrow \infty} ||\boldsymbol{g}_k|| = 0 \tag{4.67}$$

ou

$$\sum_{k=0}^{\infty} \frac{||\boldsymbol{g}_k||^4}{||\boldsymbol{d}_k||^2} < \infty. \tag{4.68}$$

Demonstração. Para provar tal proposição é suficiente verificar que se a Equação (4.67) é falsa, então a Equação (4.68) é verdadeira. Para tanto, suponha que existe $\gamma > 0$ tal que

$$||\boldsymbol{g}_k|| \geq \gamma. \tag{4.69}$$

Para um $k \geq 0$, usando a Equação (4.50) e o Teorema 4.2.1, obtém-se

$$\begin{aligned}\frac{||\boldsymbol{d}_k||^2}{||\boldsymbol{d}_{k-1}||^2} &= \frac{(\rho_{k-1}\beta_k)^2||\boldsymbol{d}_{k-1}||^2 + \theta_k^2||\boldsymbol{g}_k||^2 - 2\theta_k \boldsymbol{d}_k^T \boldsymbol{g}_k}{||\boldsymbol{d}_{k-1}||^2} \\ &\geq (\rho_{k-1}\beta_k)^2 + \theta_k^2 \frac{||\boldsymbol{g}_k||^2}{||\boldsymbol{d}_{k-1}||^2},\end{aligned} \tag{4.70}$$

pois, $\boldsymbol{d}_k^T \boldsymbol{g}_k = \boldsymbol{g}_k^T \boldsymbol{d}_k < 0$, já que $\boldsymbol{d}_k$ é direção de descida. Além disso, multiplicando ambos os lados da Equação (4.50) por $\boldsymbol{g}_k^T$, tem-se que

$$\boldsymbol{g}_k^T \boldsymbol{d}_k - \rho_{k-1}\beta_k \boldsymbol{g}_k^T \boldsymbol{d}_{k-1} = -\theta_k ||\boldsymbol{g}_k||^2. \tag{4.71}$$

Dado o fato que o tamanho ótimo do passo, ρ^* é determinado por busca em linha de Wolfe, tem-se que a seguinte condição

$$|\boldsymbol{g}_{k+1}^T \boldsymbol{d}_k| \leq c|\boldsymbol{g}_k^T \boldsymbol{d}_k|, \tag{4.72}$$

deve ser satisfeita para um $0 < c < 1$. Então pelo uso da desigualdade triangular seguida pela desigualdade de Cauchy-Schwarz, na Equação (4.72), tem-se que

$$\begin{aligned}|\boldsymbol{g}_k^T \boldsymbol{d}_k| + c_2\rho_{k-1}|\beta_k|\boldsymbol{g}_{k-1}^T \boldsymbol{d}_{k-1}| &\geq \theta_k ||\boldsymbol{g}_k||^2 \\ (\boldsymbol{g}_k^T \boldsymbol{d}_k)^2 + (\rho_{k-1}\beta_k)^2 (\boldsymbol{g}_{k-1}^T \boldsymbol{d}_{k-1})^2 &\geq \frac{\theta_k^2}{1+c_2^2}||\boldsymbol{g}_k||^4.\end{aligned} \tag{4.73}$$

Pelo uso das equações (4.70) e (4.84), obtem-se que

$$\begin{aligned}\frac{(\boldsymbol{g}_k^T \boldsymbol{d}_k)^2}{||\boldsymbol{d}_k||^2} + \frac{(\boldsymbol{g}_{k+1}^T \boldsymbol{d}_{k-1})^2}{||\boldsymbol{d}_{k-1}||^2} &= \frac{1}{||\boldsymbol{d}_k||^2}\left[(\boldsymbol{g}_k^T \boldsymbol{d}_k)^2 + \frac{||\boldsymbol{d}_k||^2}{||\boldsymbol{d}_{k-1}||^2}(\boldsymbol{g}_{k+1}^T \boldsymbol{d}_{k-1})^2\right] \\ &\geq \frac{1}{||\boldsymbol{d}_k||^2}\left[\frac{\theta_k^2}{1+c^2}||\boldsymbol{g}_k||^4 + (\boldsymbol{g}_{k+1}^T \boldsymbol{d}_{k-1})^2\left(\frac{||\boldsymbol{d}_k||^2}{||\boldsymbol{d}_{k-1}||^2} - (\rho_{k-1}\beta_k)^2\right)\right] \\ &\geq \frac{||\boldsymbol{g}_k||^4}{||\boldsymbol{d}_k||^2}\left[\frac{\theta_k^2}{1+c^2} - \theta_k^2 \frac{(\boldsymbol{g}_{k+1}^T \boldsymbol{d}_{k-1})^2}{||\boldsymbol{d}_{k-1}||^2}\frac{1}{||\boldsymbol{g}_k||^2}\right] \\ &= \frac{||\boldsymbol{g}_k||^4}{||\boldsymbol{d}_k||^2}\theta_k^2\left[\frac{1}{1+c^2} - \frac{(\boldsymbol{g}_{k+1}^T \boldsymbol{d}_{k-1})^2}{||\boldsymbol{d}_{k-1}||^2}\frac{1}{||\boldsymbol{g}_k||^2}\right].\end{aligned} \tag{4.74}$$

Segue-se da proposição 4.3 que

$$\lim_{k \longrightarrow \infty} \frac{(\boldsymbol{g}_{k-1}^T \boldsymbol{d}_{k-1})^2}{||\boldsymbol{d}_{k-1}||^2} = 0. \tag{4.75}$$

Pelo uso da Equação (4.69) e do fato que $\theta_k \geq m$, para todo k suficientemente grande, então existe uma constante λ tal que

$$\theta_k^2\left[\frac{1}{1+c_2^2} - \frac{(\boldsymbol{g}_{k-1}^T \boldsymbol{d}_{k-1})^2}{||\boldsymbol{d}_{k-1}||^2}\frac{1}{||\boldsymbol{g}_k||^2}\right] \geq \lambda. \tag{4.76}$$

Finalmente, das equações (4.74) e (4.76), segue-se que

$$\frac{(\boldsymbol{g}_k^T \boldsymbol{d}_k)^2}{||\boldsymbol{d}_k||^2} + \frac{(\boldsymbol{g}_{k-1}^T \boldsymbol{d}_{k-1})^2}{||\boldsymbol{d}_{k-1}||^2} \geq \lambda \frac{||\boldsymbol{g}_k||^4}{||\boldsymbol{d}_k||^2}. \tag{4.77}$$

Junto com a condição de Zoutendijk, chega-se a expressão de interesse mostrada na Equação (4.68). □

Uma consequência imediata da proposição 4.4 é que se

$$\sum_{k=0}^{\infty} \frac{1}{||\boldsymbol{d}_k||^2} = \infty, \tag{4.78}$$

então

$$\liminf_{k \longrightarrow \infty} ||\boldsymbol{g}_k|| = 0. \tag{4.79}$$

Utilizando todos os resultados apresentados anteriormente é possível estabelecer o teorema de convergência global do algoritmo SCG-LSSVM.

Teorema 4.2.2. *Caso exista uma constante $\gamma > 0$ tal que $||\boldsymbol{g}_k|| \geq \gamma$, então a sequência gerada pelo modelo SCG-LSSVM $\{\boldsymbol{\alpha}_k\}$ satisfaz a Equação (4.79).*

Demonstração. Do teorema 4.2.1, segue-se que

$$\boldsymbol{g}_{k-1}^T \boldsymbol{s}_{k-1} \leq -c||\boldsymbol{g}_{k-1}||\,||\boldsymbol{s}_{k-1}||. \tag{4.80}$$

Deve-se notar ainda que $\boldsymbol{l}_{k-1}^T \boldsymbol{s}_{k-1} = \boldsymbol{g}_k^T \boldsymbol{s}_{k-1} - \boldsymbol{g}_{k-1}^T \boldsymbol{s}_{k-1} \geq (c_2 - 1)\boldsymbol{g}_{k-1}^T \boldsymbol{s}_{k-1}$ então

$$\boldsymbol{l}_{k-1}^T \boldsymbol{s}_{k-1} \geq c(1-c_2)||\boldsymbol{g}_{k-1}||\,||\boldsymbol{s}_{k-1}||. \tag{4.81}$$

Além disso, das equações (4.59), (4.61), e (4.65), segue-se que

$$\begin{aligned} \beta_k \leq M \frac{||\boldsymbol{g}_k||^2}{\boldsymbol{l}_{k-1}^T \boldsymbol{s}_{k-1}} &\leq \frac{M}{c(c_2-1)} \frac{||\boldsymbol{g}_k||^2}{||\boldsymbol{g}_{k-1}||\,||\boldsymbol{s}_{k-1}||} \\ &\leq \frac{M\Gamma^2}{c\gamma(c_2-1)} \frac{1}{||\boldsymbol{s}_{k-1}||} = \frac{\mu}{||\boldsymbol{s}_{k-1}||}, \end{aligned} \tag{4.82}$$

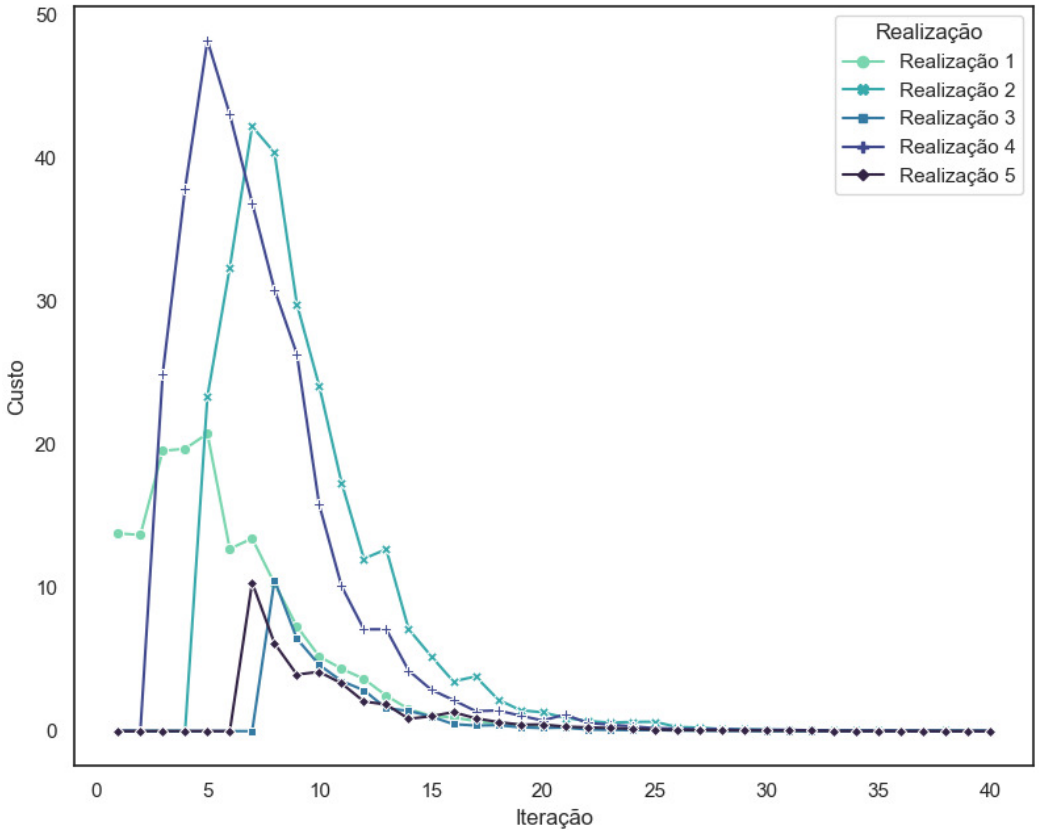
em que $\mu = \frac{M\Gamma^2}{c\gamma(c_2-1)}$ e, portanto,

$$||\boldsymbol{d}_k|| \leq |\theta_k||\boldsymbol{g}_k|| + |\beta_k|\,||\boldsymbol{s}_{k-1}|| \leq M\Gamma + \mu. \tag{4.83}$$

Isto implica que $\sum_{k=0}^{\infty} \frac{1}{||\boldsymbol{d}_k||^2} = \infty$ e pela Equação (4.78) chega-se ao resultado de interesse apresentado na Equação (4.79). □

Para verificar a capacidade de convergência do algoritmo SCG-LSSVM, gráficos da função de perda, $\mathscr{D}$, foram desenvolvidos em função das iterações para 5 realizações, utilizando conjuntos de dados sintéticos obtidos a partir da amostragem de uma distribuição normal, conforme indicado na Figura 7.

Figura 7 – Gráficos de convergência para o algoritmo SCG.



Fonte: Elaborada pelo autor.

4.2.5 Complexidade

Para o algoritmo SCG-LSSVM, tem-se que a complexidade é dominada pelo cálculo de atualização do gradiente, dado pela expressão $\boldsymbol{g}(\boldsymbol{\alpha}_k) = \tilde{\boldsymbol{K}}\boldsymbol{\alpha}_k - \boldsymbol{y}$ onde o produto da matriz de *kernel* regularizada com o vetor de multiplicadores de Lagrange denso apresenta custo computacional de $2n^2$. Além disso, as demais operações realizadas para o algoritmo apresentam complexidade linear $O(n)$. Assim, como estes termos tendem a se tornar insignificantes com o aumento do número de amostras n, tem-se que a complexidade final para o algoritmo por iteração é quadrática $O(n^2)$. Um resumo da contribuição para a complexidade dos principais passos do algoritmo SCG-LSSVM é indicado na Tabela 4.

Vale destacar, que apesar da utilização de informação de segunda ordem (dada pela

Tabela 4 – Resumo da complexidade por iteração.

Operação	Flops
Gradiente $\tilde{\boldsymbol{K}}\boldsymbol{\alpha}_k - \boldsymbol{y}$	$2n^2$
Direção de descida $\boldsymbol{d}_k$	$3n$
Atualização BFGS *memoryless*	$7n$
Passo ótimo ρ_k^*	$8n$
Atualização $\boldsymbol{\alpha}_{k+1}$	$2n$
Total por iteração	$2n^2 + 20n$

Fonte: Elaborada pelo autor.

Hessiana da função dual) o que poderia contribuir para uma complexidade cúbica $O(n^3)$ para os esquemas de atualização, evita-se tal custo com o emprego da aproximação BFGS, resultando em uma metodologia mais eficiente por iteração. Além disso, o emprego da aproximação da Hessiana permite alto ganho funcional, tendo como consequência a convergência em um menor número de iterações.

Obviamente, para cenários com restrições de memória, técnicas de fatorização e alocação dos valores das colunas em memória (cache) para a matriz de *kernel* podem ser adotadas para reduzir o custo de armazenamento sem aumentar o custo computacional.

4.2.6 Poda Iterativa

Assim como o algoritmo TCSMO, o SCG aplicado diretamente na solução do problema dual do treinamento de LSSVMs, fornece um vetor ótimo de multiplicadores de Lagrange denso, o que impacta negativamente no estágio de predição para o processamento de grandes bases de dados. Neste sentido, uma técnica de poda iterativa baseada na distância de um padrão ao hiperplano de decisão foi adotada. Sabe-se que a função de predição para o modelo LSSVM em problemas de classificação binária é dada como se segue

$$f(\boldsymbol{x}) = sinal(g(\boldsymbol{x})), \tag{4.84}$$

em que,

$$f(x) = sinal(x) = \begin{cases} +1, \ \ se \ x \geq 0 \\ -1, \ \ \text{caso contrário} \end{cases} \tag{4.85}$$

e

$$g(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b. \tag{4.86}$$

Sendo $\boldsymbol{x}$ o padrão a ser classificado, $\boldsymbol{x}_i$ o i-ésimo padrão de treinamento e $K(\boldsymbol{x}_i,\boldsymbol{x}) = \phi(\boldsymbol{x})^T\phi(\boldsymbol{x}_i)$ indicando o *kernel trick*. Destaca-se que a expressão dada abaixo

$$d(\boldsymbol{x}_i) = \frac{|g(\boldsymbol{x}_i)|}{||\boldsymbol{w}||}, \tag{4.87}$$

representa a distância de um padrão $\boldsymbol{x}_i$ ao hiperplano dado pelos valores dos pesos contidos no vetor $\boldsymbol{w}$. Desta forma, é possível considerar $|g(\boldsymbol{x})|$ um tipo de medida de distância, já que o vetor $\boldsymbol{w}$ permanece o mesmo para todos os padrões de treinamento.

Vale destacar, por meio da análise da função de predição apresentada na Equação (4.86), que a mesma pode ser vista como a média ponderada dos vetores-suporte de cada classe. Considerando o problema binário com **kernel** linear, tem-se que $y \in \{-1,+1\}$ e com isso, tem-se que

$$g(\boldsymbol{x}) = \left(\sum_{i\in SV^+} \alpha_i\boldsymbol{x}_i - \sum_{j\in SV^-} \alpha_j\boldsymbol{x}_j\right)\boldsymbol{x} + b, \tag{4.88}$$

logo,

$$g(\boldsymbol{x}) = (\boldsymbol{\mu}_i^T - \boldsymbol{\mu}_j^T)\boldsymbol{x} + b, \tag{4.89}$$

em que, $\boldsymbol{\mu}_i$ e $\boldsymbol{\mu}_j$ representam os vetores média ponderada para as classes positiva e negativa, respectivamente. SV^+ e SV^- indicam os conjuntos de índices dos vetores-suporte da classe positiva e negativa, respectivamente. Com isso, ao se considerar novos vetores-suportes e a depender de sua classe, o mesmo contribuirá para variações nos valores das médias ponderadas impactando no posicionamento do hiperplano de decisão.

A heurística empregada para a realização da poda, considera que os padrões de treinamento mais próximos ao hiperplano de decisão são aqueles mais representativos para sua formação, ou seja, apresentam maiores chances de serem vetores-suporte. A cada iteração do algoritmo SCG-LSSVM, os multiplicadores são atualizados e uma verificação de quais deles estão mais próximos do hiperplano de decisão é realizada, utilizando como critério os menores valores de $|g(\boldsymbol{x})|$.

Uma vez determinado quais os padrões mais próximos do hiperplano, estes são inseridos em um conjunto ativo e serão atualizados, enquanto os demais não passam por atualizações. Neste ponto, vale destacar que os multiplicadores são inicializados com valores nulos, portanto todos estão inicialmente no conjunto inativo.

Nota-se que com o passar das iterações, mais multiplicadores de Lagrange serão diferentes de zero e com isso mais padrões serão inseridos no conjunto ativo, devido as atualizações dos multiplicadores o que ocasiona uma movimentação do hiperplano. Utilizando um

conjunto de validação e uma métrica de desempenho, como a acurácia, é possível definir um critério de parada com base na variação da métrica quando avaliado no conjunto de validação.

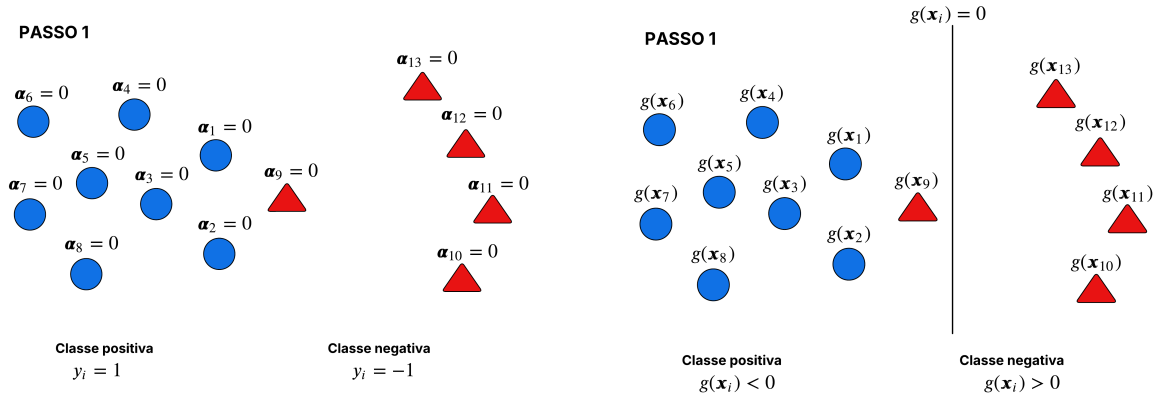
Dado que as atualizações são calculadas com base no algoritmo SCG, tem-se garantia de convergência desse procedimento de poda. Ao final do processo, obtém-se uma solução ótima esparsa com base nos multiplicadores não nulos, além disso, vale frisar que diferentemente de outros métodos de poda, esta heurística baseia-se no crescimento do conjunto ativo e, com isso, não é necessário definir um percentual de poda. Assim, este método é de tamanho variável o que contribui para a redução do número de hiperparâmetros a serem otimizados. Para melhor detalhar o comportamento dos multiplicadores de Lagrange ao considerar tal metodologia de poda, as Figuras 8 e 9 apresentam o passo da alteração dos valores dos multiplicadores, bem como, a movimentação correspondente do hiperplano de separação. O Algoritmo 10 ilustra a adição do procedimento de poda iterativa no algoritmo SCG.

O algoritmo é similar ao Algoritmo 9, as diferenças são observadas entre as linhas 7 e 21. A solução inicial dos multiplicadores de Lagrange é o vetor nulo, então todos os padrões de treino estão no conjunto inativo inicialmente. Após a primeira atualização alguns multiplicadores terão valores diferentes de zero, para estes, calcule a distância ao hiperplano como dado pelo termo $|g(\boldsymbol{x}_i)|$ e indicado na linha 7.

Em seguida, selecione os N_r padrões com menores valores de $|g(\boldsymbol{x}_i)|$, estes estarão no conjunto ativo a partir de então. Com o passar das iterações mais multiplicadores terão valores diferentes de zero e novos padrões, a depender de sua proximidade ao hiperplano, entrarão no conjunto ativo. Em cada iteração, calcula-se a acurácia do SCG-LSSVM considerando apenas os padrões do conjunto ativo e se verifica a variação da métrica em duas iterações sucessivas. Caso a variação, fique abaixo de um dado limiar positivo, então significa que o algoritmo estabilizou e que tem-se uma solução final, possivelmente esparsa. Estes cenários são ilustrados das linhas 11 à 22, o restante do algoritmo é análogo ao Algoritmo 9.

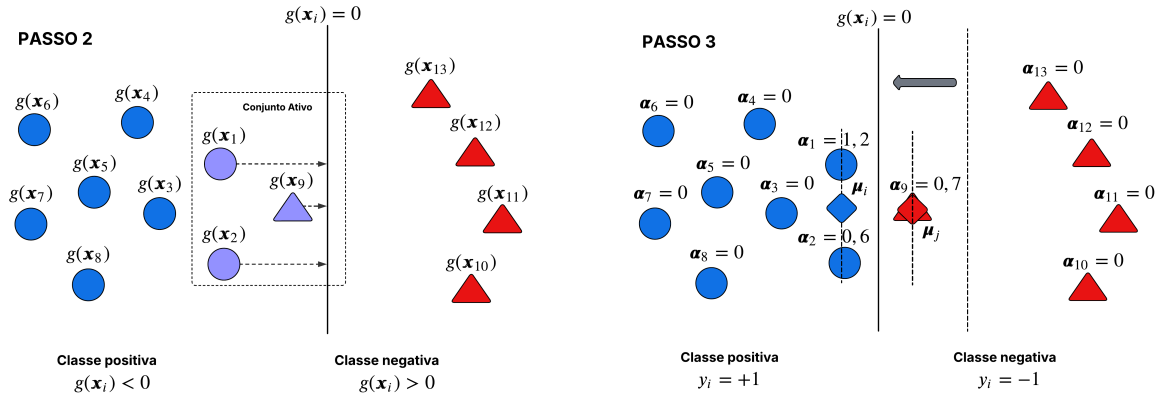
Neste capítulo as duas propostas denominadas TCSMO-LSSVM e SCG-LSSVM são apresentadas ilustrando as motivações para o seu uso, detalhamento matemático, análise de convergência e complexidade para ambas. Também são apresentados ainda que de forma superficial as metodologias precursoras de cada proposta, bem como, os principais avanços teóricos que as propostas fornecem quando comparados a este métodos iniciais. No próximo capítulo, apresenta-se os materiais, que são representados pelas bases de dados, empregados na avaliação das duas novas propostas tanto em cenários de classificação binária como de

Figura 8 – Representação do procedimento de poda: PASSO 1



Fonte: Elaborada pelo autor.

Figura 9 – Representação do procedimento de poda: PASSO 2 e PASSO 3



Fonte: Elaborada pelo autor.

aproximação de funções. Procedimentos de pré-processamento destas bases e otimização de hiperparâmetros dos modelos avaliados ainda são detalhados. Por fim, apresenta-se o estudo de caso associado a base de dados de FOLSOM para previsão de irradiância solar de curto prazo, em que as propostas são avaliadas em termos de previsão de séries temporais.

Algoritmo 10: SCG-LSSVM com poda baseada na distância $g(\boldsymbol{x})$

Entrada: $\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}_0, \varepsilon, N_r, \nu$ e $1 \leq \phi \leq 2$

início

  Inicie $k = 0$, $\boldsymbol{\alpha}_0 = \mathbf{0}$, $Ativo = \varnothing$ e $Inativo = \{1, 2, \ldots, N\}$;

  Calcule $f(\boldsymbol{\alpha}_0) = f_0 = \frac{1}{2}\boldsymbol{\alpha}_0^T \tilde{\boldsymbol{K}} \boldsymbol{\alpha}_0 - \boldsymbol{y}^T \boldsymbol{\alpha}_0$ e $\boldsymbol{g}_0 = \nabla f(\boldsymbol{\alpha}_0) = \tilde{\boldsymbol{K}} \boldsymbol{\alpha}_0 - \boldsymbol{y}$;

  while $||\boldsymbol{g}_k|| \leq \varepsilon$ **do**

    Calcule o tamanho do passo $\boldsymbol{\rho}_k$ usando a busca linear como dado na Equação (4.57);

    Atualize a variável dual, fazendo $\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \rho_k^* \boldsymbol{d}_k$ e o gradiente $\boldsymbol{g}_{k+1}$;

    Determine o valor de $g(\boldsymbol{x})$ para cada padrão de treinamento

    Selecione os N_r padrões mais próximos ao hiperplano de decisão

    Atualize os conjuntos ativo e inativo;

    Atualize os multiplicadores e o gradiente para os padrões no conjunto ativo;

    Calcule as acurácias do modelo, Acc e $Acc_{anterior}$, no conjunto de validação;

    if $(Acc - Acc_{anterior}) > 0$ **then**

      if $(Acc - Acc_{anterior}) < \nu$ **then**

        break

      end

      else

        pass

      end

    end

    else

      pass

    end

    Calcule os parâmetros θ_k e β_k usando a fórmula de truncamento dada pelas equações (4.59) e 4.60;

    Atualize a direção de descida $\boldsymbol{d}_{k+1}$ utilizando a Equação 4.50 e faça $k \longleftarrow k+1$. Retorne ao PASSO 5.

  end

fim

Resultado: Multiplicadores de Lagrange ótimos $\boldsymbol{\alpha}^*$

5 MATERIAIS E MÉTODOS

Neste capítulo, são discutidos os métodos e as bases de dados utilizadas para a avaliação do desempenho preditivo das duas propostas discutidas no capítulo anterior, o foco é ilustrar os conjuntos de dados de *benchmarking* considerados tanto para o cenário de classificação binária como o de aproximação de funções. Um detalhamento sobre o processo de otimização de hipeparâmetros é abordado, além de, ter-se uma avaliação das bases sintéticas para a análise empírica da qualidade da fronteira de decisão gerada pelas novas propostas.

5.1 Simulações para Classificação de Padrões

Com o intuito de validar as propostas já apresentadas nesta tese, foram conduzidas algumas análises com foco em três tópicos principais:

(i) avaliar a capacidade preditiva das propostas considerando algumas bases de dados reais e sintéticas com baixo e alto volume de dados;

(ii) avaliar de forma empírica a qualidade da fronteira de decisão obtida pelas propostas desta tese quando aplicadas em algumas bases de dados sintéticas;

(iii) avaliar o processo de ajuste de hiperparâmetros dos modelos propostos.

Para fins de reprodutibilidade, deve-se observar que as configurações de *hardware* usadas para realizar os experimentos computacionais envolveram a utilização de uma máquina com CPU Intel(R) Core(TM) i5-7400 e 16 GB de RAM.

5.1.1 Bases de Dados Pequenas

A novas propostas são validadas por meio de experimentos computacionais sobre oito conjuntos de dados reais obtidos do repositório da Universidade da Califórnia Irvine (*UCI machine learning repository*, UCI) (Bay *et al.*, 2000). Todos os experimentos envolvem a tarefa de classificação binária e as informações sobre seu tamanho e atributos são apresentadas na Tabela 5. Vale ressaltar que cada um dos conjuntos de dados foram divididos aleatoriamente em proporções de 70% e 30% para treinamento e teste, respectivamente.

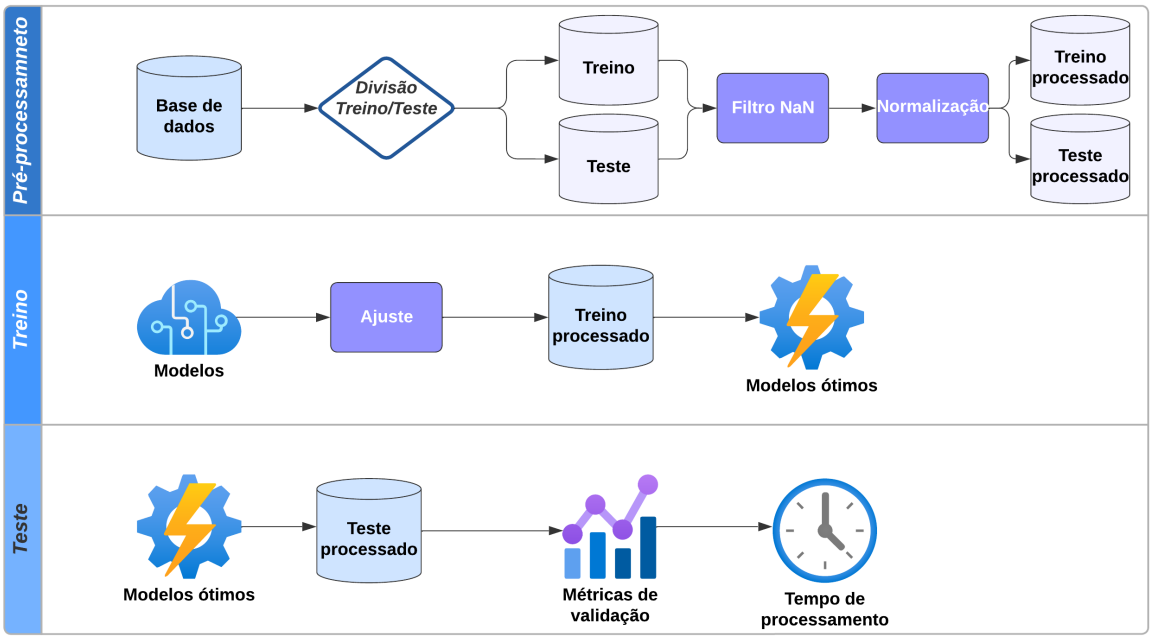
Ambas as parcelas passaram por um fluxo de dados padrão, em que realizou-se a remoção de valores nulos e também a realização de normalização estatística fazendo com que as variáveis fiquem com desvio padrão unitário e média nula, evitando assim o viés associado ao vazamento de dados (Géron, 2022). O fluxo completo é representado na Figura 10

Tabela 5 – Detalhamento sobre as bases de dados consideradas.

Base	Abreviação	Tamanho	Atributos
Haberman	HAB	306	3
Prima Indian Diabet	PID	768	8
Breast Cancer Wisconsin	BCW	570	10
Vertebral Column Pathologies	VCP	310	6
Bupa Liver Disorders	BLD	345	5
Ionosphere	ION	351	34
Statlog German Credit	GCR	1000	20
Statlog Australian Credit	AUS	690	14

Fonte: Elaborada pelo autor.

Figura 10 – *Pipeline* de dados completo utilizado para obtenção dos resultados.



Fonte: Elaborada pelo autor.

5.1.2 Ajuste de Hiperparâmetros

Otimização Bayesiana foi empregada para a realização do ajuste de todos os modelos (Frazier, 2018), este procedimento é menos oneroso em termos computacionais quando comparado aos tradicionais métodos de busca de hiperparâmetros como busca em grade e busca aleatória (Géron, 2022).

A busca é realizada em determinados pontos selecionados aleatoriamente e uma métrica de desempenho é avaliada em cada cenário, uma vez que ocorre uma redução na capacidade preditiva, novos pontos são selecionados em uma vizinhança distinta daqueles em que houve piora, fazendo com que a avaliação não continue em uma direção de queda de acurácia.

Vale ressaltar que o *kernel* utilizado neste trabalho foi o radial gaussiano e que os

hiperparâmetros selecionados para ajuste foram o termo σ da função de *kernel* e o termo de regularização τ do problema primal do LSSVM. A Tabela 6 apresenta o espaço de busca para cada um deles. Neste trabalho, a etapa de otimização bayesiana utilizou 150 experimentos independentes.

Tabela 6 – Espaço de busca para cada hiperparâmetro.

Hiperparâmetro	**Espaço de busca**	**Escala**
σ	$[10^{-2}, 10^{2}]$	Log
τ	$[10^{-2}, 10^{2}]$	Log

Fonte: Elaborada pelo autor.

5.1.3 Fronteira de Decisão

Para investigar a qualidade da fronteira de decisão gerada pelo TCSMO-LSSVM e SCG-LSSVM, considera-se as bases da dados sintéticas *Two Moon* (TWM), *Two Square* (TWS) e *Two Circle* (TWC) que são comumente empregadas para validar a qualidade da fronteira de decisão desenvolvida por um dado modelo, bem como, o nível de esparsidade gerado por uma nova proposta. Além disso, considera-se o modelo IP-LSSVM, uma variante esparsa do LSSVM bem consolidada, como referência para a avaliação empírica da qualidade da fronteira. Informações sobre as bases de dados sintéticas são reportadas na Tabela 7.

Tabela 7 – Bases de dados sintéticas utilizadas.

Base	**Abreviação**	**Tamanho**	**Atributos**
Artificial 1 (*Two Squares*)	TWS	1000	2
Artificial 2 (*Two Moon*)	TWM	800	2
Artificial 3 (*Two Circle*)	TWC	800	2

Fonte: Elaborada pelo autor.

5.1.4 Bases de Dados Grandes

Com o intuito de também avaliar as propostas em termos de capacidade preditiva e velocidade de treinamento em grandes bases de dados, o TCSMO-LSSVM e o SCG-LSSVM foram avaliados em três grandes bases de referência. A Tabela 8 apresenta informações sobre tamanho e dimensionalidade de cada base.

Tabela 8 – Grandes bases de dados consideradas.

Base	Abreviação	Tamanho	Atributos
ADULT	ADT	48,842	123
SHUTTLE	SHT	58,000	9
Bank Marketing	BKM	4,521	17

Fonte: Elaborada pelo autor.

5.2 Simulações para Aproximações de Funções

Nesta seção, realiza-se a avaliação do desempenho das duas propostas em tarefas de aproximações de funções. Para tanto, foram realizadas diversas simulações computacionais em quatro bases de dados de reais e sintéticas, além de, também ter sido considerado uma base de dados de referência envolvendo o problema de previsão de curto prazo de irradiância solar. O detalhamento sobre o processo de coleta, estatísticas descritivas e pré-processamento para esta base é apresentada na próxima subseção. O foco da avaliação está nos seguintes tópicos principais:

(i) avaliar a capacidade preditiva das propostas em termos das métricas raiz do erro quadrático médio (RMSE) e coeficiente de determinação (R^2), considerando algumas bases de dados reais e sintéticas;

(ii) avaliar o tempo de processamento para o treinamento das propostas frente aos dos demais modelos de comparação;

(iii) avaliar o nível de esparsidade obtido para cada proposta, bem como, o impacto sobre a performance dado um determinado de nível de esparsidade; e

(iv) avaliar o processo de ajuste de hiperparâmetros dos modelos propostos.

5.2.1 Bases de Dados Consideradas

As propostas são validadas por meio de experimentos numéricos em quatro conjuntos de dados reais obtidos do repositório da Universidade da Califórnia Irvine (*UCI machine learning repository*, UCI) (Bay *et al.*, 2000). Todos eles envolvem a tarefa de regressão. Informações sobre seu tamanho e quantidade de atributos são apresentadas na Tabela 9. Vale ressaltar que todos os conjuntos de dados foram divididos aleatoriamente em proporções de 70% e 30% para treinamento e teste, respectivamente.

Ambas as parcelas passaram por um *pipeline* de dados padrão, onde realizou-se a remoção de valores nulos e também a realização de normalização estatística fazendo com que as

variáveis fiquem com desvio padrão unitário e média nula, assim como foi realizado para o caso de classificação binária. O fluxo de dados completo é o mesmo apresentado para classificação, mostrado na Figura 10.

Tabela 9 – Detalhamento sobre as bases de dados consideradas para o problema de regressão.

Base	Abreviação	Tamanho	Atributos
Abalone	ABA	4176	9
AutoMPG	MPG	397	9
Concrete	CON	1030	9
Energy	ENE	768	10

Fonte: Elaborada pelo autor.

5.2.2 Ajuste de Hiperparâmetros

Assim como no caso de classificação binária, o procedimento de ajuste dos modelos foi realizado pelo uso de otimização Bayesiana, entretanto, a métrica de desempenho utilizada para este cenário foi o coeficiente de determinação R^2, enquanto o número de experimentos independentes foi superior ao realizado para o caso de classificação tendo sido utilizado 200 experimentos, o que é justificado devido a um aumento no espaço de busca, como indicado na Tabela 10.

Tabela 10 – Espaço de busca para cada hiperparâmetro.

Hiperparâmetro	Espaço de busca	Escala
σ	$[10^{-4}, 10^{4}]$	Log
τ	$[10^{-4}, 10^{4}]$	Log

Fonte: Elaborada pelo autor.

5.3 Estudo de Caso: Base de FOLSOM, CA

Sendo um dos objetivos avaliar o desempenho das duas propostas em um contexto de previsão de séries temporais não lineares, foi considerado a base de irradiância solar de Folsom, Califórnia. O comportamento intermitente e estocástico da fonte solar torna o problema de alta complexidade, além disso, a alta volumetria da base permite uma avaliação da eficiência dos algoritmos tanto em termos de tempo como de memória.

Os dados foram coletados a partir de uma série de sensores de irradiância solar e meteorológicos situados em locais da Costa Oeste dos Estados Unidos. Os sensores de

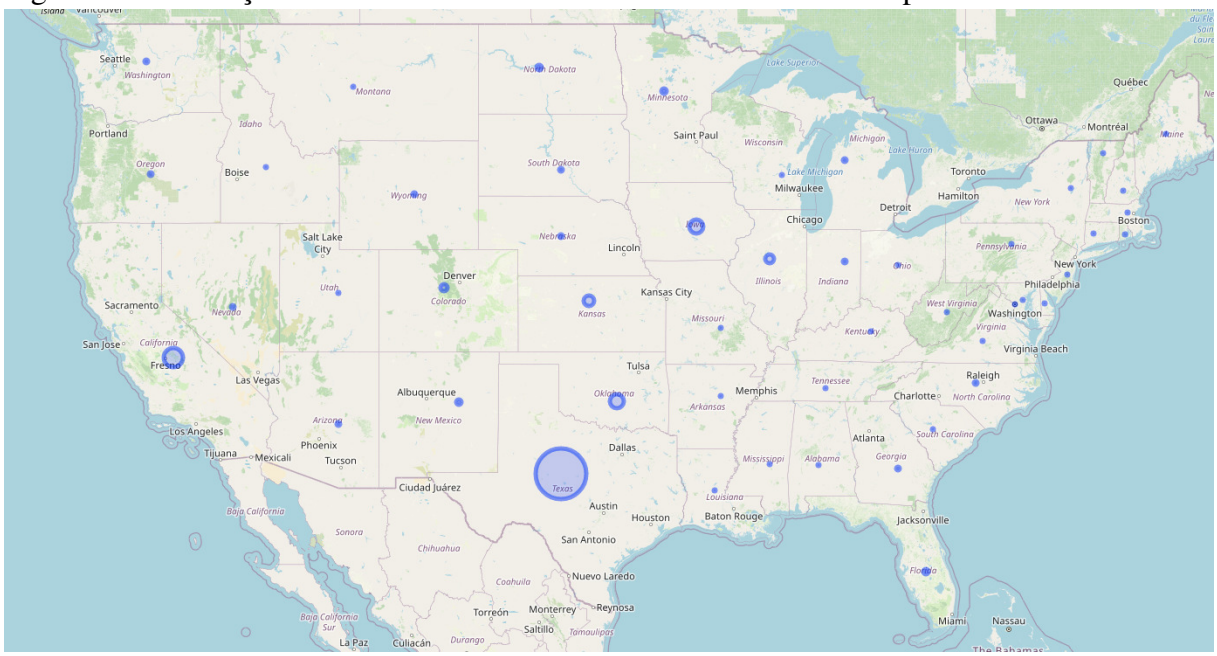
irradiância mediram tanto irradiância horizontal global (*global horizontal irradiance*, GHI), quanto irradiância normal direta (*direct normal irradiance*, DNI), com taxas de amostragem de 1 minuto. Câmeras com lentes 'olho de peixe' foram utilizadas para fornecer imagens do céu.

A escolha por Folsom, Califórnia, baseou-se no fato de que os dados deveriam ser provenientes de áreas de interesse para previsão solar, ou seja, áreas com altos níveis de geração de energia solar preexistente ou planejada. A Figura 11 mostra a produção de eletricidade obtida a partir de fontes eólicas e solares, onde se pode observar que a Califórnia ocupa o segundo lugar no uso de fontes de energia renováveis, sendo superada apenas pelo Estado do Texas.

O segundo motivo é que os dados devem abranger dois ou mais anos sequencialmente, para que os conjuntos de treinamento e teste tenham pelo menos um ano de duração cada. Terceiro, todas as fontes de dados para o local devem ser de alta qualidade, com intervalos mínimos de dados ausentes e com alto nível de controle de qualidade. Em quarto lugar, os dados devem incluir as entradas exógenas mais comuns para a previsão solar, como imagens do céu, imagens de satélite e previsão climática numérica (*numerical weather prediction*, NWP), (Pedro *et al.*, 2019).

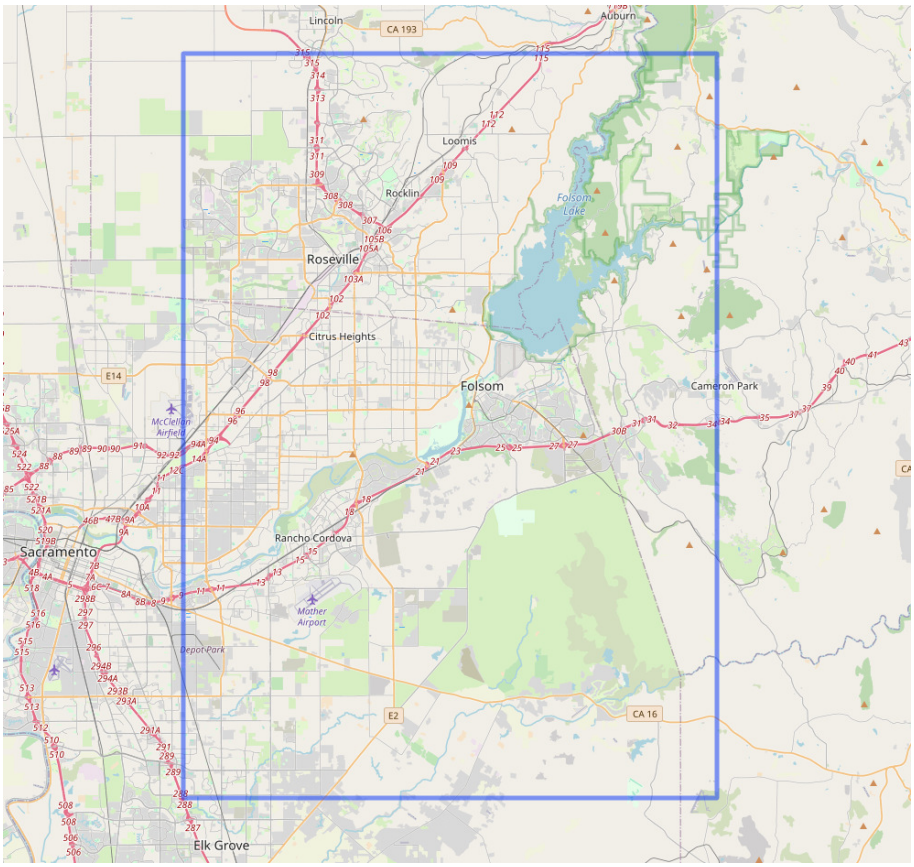
Folsom (38.642°, -121.148°) é uma cidade no Condado de Sacramento, no Vale Central da Califórnia, como indicado pela Figura 12, com uma classificação Csa (C = clima temperado; s = verão seco; a = verão quente) no esquema climático de Köppen.

Figura 11 – Produção de eletricidade drivada das fontes solar e eólica para os Estados Unidos.



Fonte: Elaborada pelo autor.

Figura 12 – Cidade de Folsom, CA.



Fonte: Elaborada pelo autor.

Imagens do céu foram capturadas à taxa de uma imagem por minuto com uma câmera de segurança (Vivotek, modelo FE8171V) apontada para o zênite no mesmo período. As vantagens dessa câmera incluem alta resolução, fácil instalação, baixo custo e ausência de partes móveis.

Por outro lado, como a luz solar direta não é bloqueada, a região circunsolar é afetada por brilho causado por espalhamento Mie para frente e também por luz espalhada pela cúpula do céu. A câmera fornece imagens JPEG compactadas de 24 *bits*, com 8 bits por canal de cor (Vermelho, Verde e Azul), em resolução de 1563 × 1538 *pixels*. Após a remoção dos *pixels* que não correspondem à cúpula celeste e dos *pixels* saturados, apenas cerca de 50% dos *pixels* são utilizáveis (Pedro *et al.*, 2019).

5.3.0.1 Dados de Irradiância

Os dados de irradiância foram médias em janelas de 5 minutos e, em seguida, as séries temporais resultantes foram normalizadas pelo índice de céu claro (k_t) para remover variações determinísticas diárias e sazonais nos dados. O k_t é definido pela razão entre o valor de

irradiância (GHI ou DNI), I, e a respectiva irradiância em condições de céu claro, I_{cs}, conforme a expressão a seguir

$$k_t = \frac{I}{I_{cs}}. \tag{5.1}$$

O modelo de céu claro utilizado para calcular k_t é o conhecido modelo de Ineichen e Perez (Ineichen e Perez, 2002). Após o cálculo de k_t para as séries temporais de GHI e DNI, três atributos foram calculados usando funções recursivas sobre os valores de k_t dentro de uma janela de processamento que antecede o tempo de emissão da previsão t.

Média Reversa (Backward Average) da série temporal do índice de céu claro: para um instante de tempo t determinado, esta característica é dada pelo vetor **B(t)** com componentes dadas por

$$B_i(t) = \frac{1}{N} \sum_{t \in]t-i\delta-T, t-T]} k_t(t), \quad i = [1,2,\ldots,M]. \tag{5.2}$$

Média Defasada (Lagged Average) para a série temporal de k_t: este atributo é representado pelo vetor $\mathbf{L}(t)$, cujos componentes são calculados utilizando a seguinte expressão

$$L_i(t) = \frac{1}{N} \sum_{t \in]t-i\delta-T, t-(i-1)\delta-T]} k_t(t), \quad i = [1,2,\ldots,M]. \tag{5.3}$$

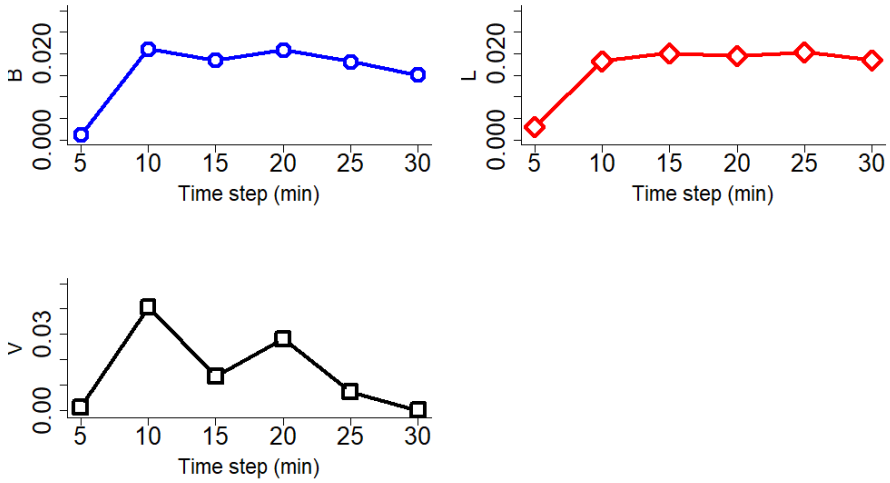
Variabilidade do k_t (Variability of k_t): Este preditor é representado pelo vetor $\mathbf{V}(t)$, cujas componentes são dadas por

$$V_i(t) = \sqrt{\frac{1}{N} \sum_{t \in]t-i\delta-T, t-T]} \Delta k_t(t)^2}, \quad i = [1,2,\ldots,M]. \tag{5.4}$$

Nessas equações, δ é o tamanho mínimo da janela, N é o número de pontos de dados na janela de processamento, $t-T$ é a extremidade direita da janela de processamento, e M é o número de janelas de processamento consideradas. Os parâmetros δ, T e M dependem do horizonte de previsão: $\delta = [5,10,15,20,25,30]$ minutos, $T = [0,0,8]$ horas, e $M = [6,6,12]$ para as previsões intra-hora, intra-dia e diária, respectivamente (Pedro e Coimbra, 2015).

Neste trabalho, foram utilizados $\delta = [5,10,15,20,25,30]$, que indicam os horizontes de previsão considerado e $T = 0$. Como o tamanho total da janela varia de 5 a 30 minutos em etapas de 5 minutos, o comprimento do recurso é $M = 6$. A Figura 13 permite uma visualização mais clara da variação desses preditores para o caso de DNI.

Figura 13 – Valores de L, B e V no conjunto de treinamento em função do passo de tempo, δ, para $\delta = [5, 10, 15, 20, 25, 30]$ min, $T = 0$ e $M = 6$.



Fonte: Elaborada pelo autor.

5.3.0.2 *Imagens do Céu*

Os atributos estatísticos extraídos das imagens do céu são a média aritmética, desvio padrão e entropia de Shannon indicadas pelas Equações (5.5), (5.6) e (5.7). Estas métricas são calculadas para cada canal das imagens no formato RGB, além da razão vermelho-para-azul ρ, cujos componentes são definidos por $\rho_i = \frac{r_i}{b_i}$ e são dadas por

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{5.5}$$

$$\sigma = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{5.6}$$

e

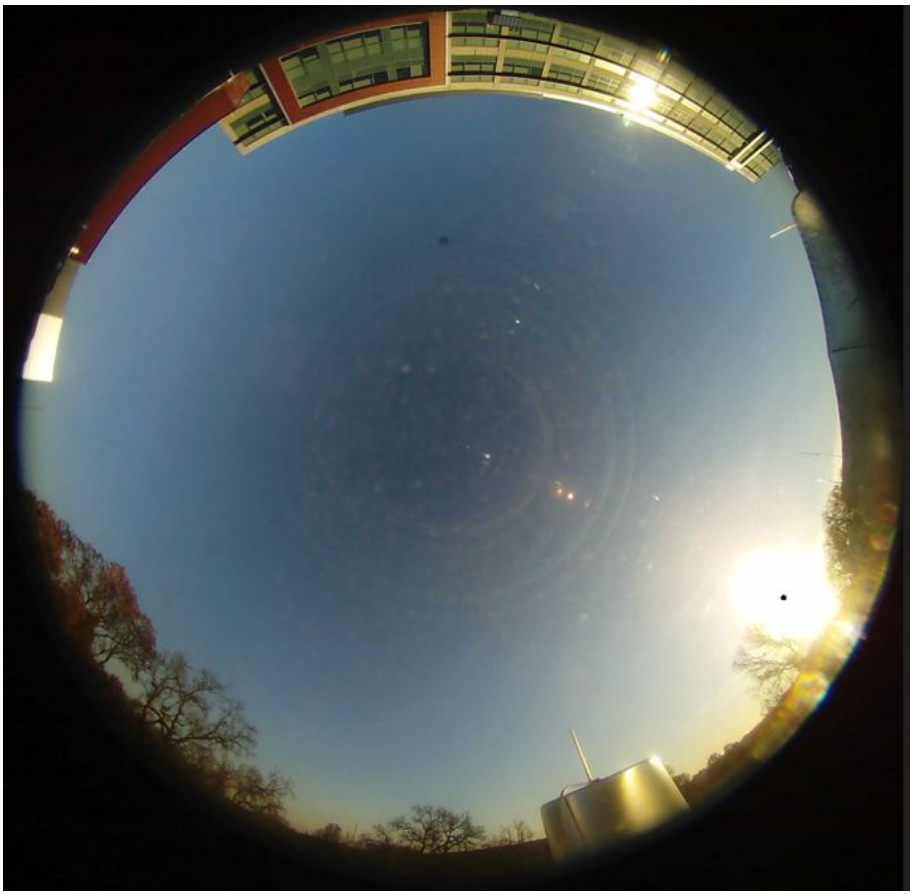
$$H(x) = -\sum_{i=0}^{255} p(x_i)\log(p(x_i)). \tag{5.7}$$

Onde x_i é a intensidade do nível de cinza para o i-ésimo pixel e $p(x_i)$ é a frequência de ocorrência do i-ésimo nível de cinza. A Figura 15 fornece um exemplo de uma imagem do céu usada para obter este banco de dados.

5.3.0.3 *Modelos para a Análise Comparativa*

O gerenciamento de dados pelo método de grupo (*group method handling of data*, GMDH) é uma família de algoritmos indutivos para modelagem matemática baseada no cômputo

Figura 14 – Exemplo de uma imagem do céu capturada em 31/12/2013 às 16h utilizada neste trabalho.



Fonte: Elaborada pelo autor.

de conjuntos de dados multiparamétricos, que apresenta otimização estrutural e paramétrica totalmente automática de modelos (Anastasakis e Mort, 2001). Os algoritmos GMDH caracterizam-se por um procedimento indutivo que realiza a seleção progressiva de modelos polinomiais gradualmente mais complexos, escolhendo a melhor solução por meio de um critério externo (Madala, 2019).

Para encontrar a melhor solução, os algoritmos GMDH consideram vários subconjuntos de componentes da função base, chamados de modelos parciais. Os coeficientes desses modelos são estimados pelo método dos mínimos quadrados. Os algoritmos GMDH aumentam gradualmente o número de componentes dos modelos parciais e encontram uma estrutura de modelo com complexidade ótima, indicada pelo valor mínimo de um critério externo. Este processo é denominado auto-organização de modelos.

Vale ressaltar que a metodologia utilizada para buscar o modelo parcial ótimo no algoritmo GMDH foi uma abordagem de busca combinatória. Esta abordagem apresenta algumas vantagens em relação às redes neurais polinomiais, mas demanda considerável poder

computacional, tornando-se ineficaz para objetos com grande número de entradas. Um importante avanço do GMDH combinatorial é seu desempenho superior ao método de regressão linear quando o nível de ruído nos dados de entrada é maior que zero. O método garante que o modelo ótimo será encontrado durante a triagem exaustiva.

O algoritmo gradiente extremo intensificado (*extreme gradient boosting*, XGBoost) é baseado em um processo de agregação de árvores de decisão em um formato sequencial, onde uma determinada árvore é construída com o intuito de minimizar o erro da árvore anterior, processo esse denominado *gradient boosting*. Esta metodologia apresenta diferenciais como a capacidade de realizar regularização l_1 e l_2 combinadas (*elastic net regularization*) o que minimiza o risco de sobreajuste, bem como, induz a geração de modelos robustos à valores anômalos.

Destaca-se também, a capacidade de tratar com variáveis categóricas de forma automática, bem como, a possibilidade de processamento distribuído durante a construção das árvores, acelerando consideravelmente o treinamento. Procedimentos de poda e de *early stopping* são outros diferenciais do XGBoost na atenuação de sobrajuste. Todas essas características levaram o algoritmo XGBoost a ser um dos modelos de aprendizagem de máquina mais comumente aplicados em tarefas de classificação e regressão para dados tabulares.

Similar ao algoritmo XGBoost, a máquina de intensificação por gradiente leve (*light gradient boosting machine*, LightGBM) também emprega uma metodologia de agregação de árvores de decisão baseado em *gradient boosting* com a adição de duas novas técnicas: Amostragem de um lado baseada em gradiente (*gradient-based one side sampling*, GOSS) e agrupamento exclusivo de atributos (*exclusive feature bundling*, EFB). Essas técnicas são projetadas para melhorar significativamente a eficiência e a escalabilidade do *gradient boosting*.

O LightGBM traz as mesmas características do XGBoost: *elastic net regularization*, poda e *early stopping* também são procedimentos passíveis de serem realizados no LightGBM, além de ter a capacidade de tratar com variáveis categóricas de uma forma automática. Por fim, para conseguir um treinamento mais rápido que o XGBoost sem perda de desempenho preditivo, o LightGBM usa a abordagem denominada *leaf-wise tree growth*. Nesta abordagem, *leaf-wise*, em cada nível, apenas um dos lados da árvore fica mais profunda. Isto significa que a cada nível, tem-se uma quantidade cada vez menor de resíduos a se considerar de modo a encontrar o limiar que maximize o ganho.

5.3.0.4 Seleção de Atributos

Nesta subseção é apresentada a metodologia para a seleção dos preditores para os modelos empregados na tarefa de previsão. A primeira etapa consiste na realização de uma análise bivariada, onde foi aplicado um procedimento de *clustering* hierárquico, utilizando como métrica de dissimilaridade $1 - cor(x_i, x_j)$, sendo a correlação de *Spearman* aplicada para este cenário.

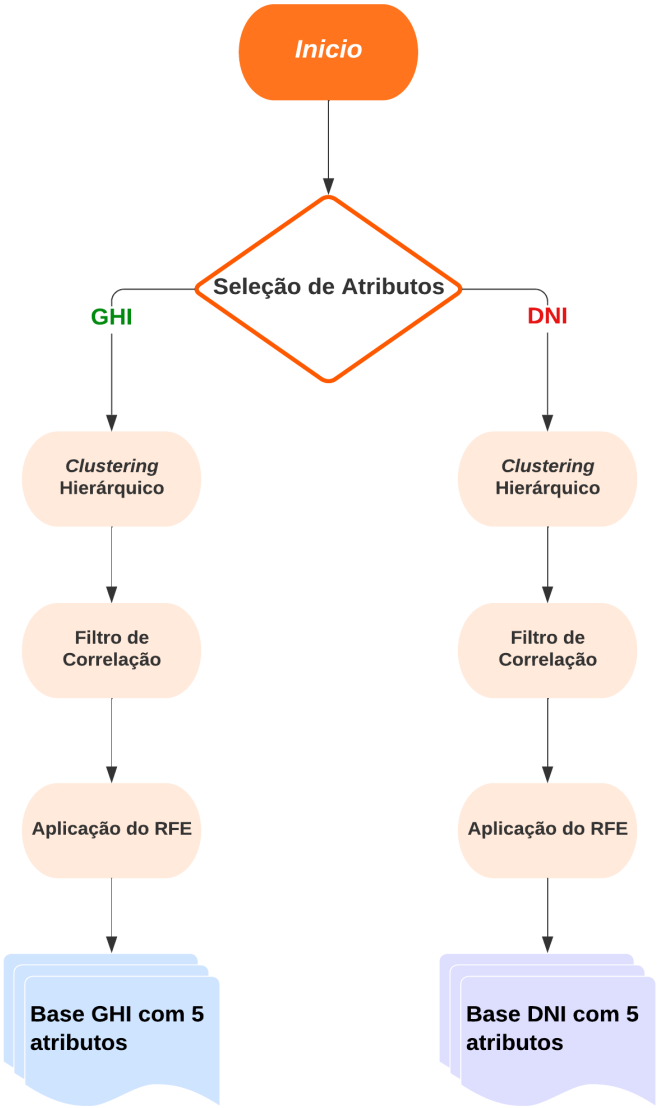
Assim, ao final dessa etapa, obtém-se um conjunto de *cluster's*, onde para cada um se tem um conjunto de variáveis correlacionadas. Em seguida, o critério para a remoção das variáveis de cada *cluster* foi verificar se a correlação com a variável alvo era superior a 0.4, desta forma evita-se problemas relacionados a multicolinearidade, sem que sejam descartadas variáveis relevantes para explicar a saída.

Após esse procedimento, as variáveis resultantes passaram pelo algoritmo de eliminação recursiva de atributos (*recursive feature elimination*, RFE) (Li e Yang, 2005), que consiste em remover variáveis menos relevantes para a performance do modelo de uma forma iterativa. Destaca-se que nesta etapa, utilizou-se o modelo XGBoost para a realização do RFE, tendo sido configurado um número final de 5 variáveis de entrada para os modelos considerados. A Figura 15 apresenta o *pipeline* de seleção de atributos.

Neste capítulo, são apresentadas as bases de dados utilizadas para a avaliação do desempenho preditivo das novas propostas, tanto no âmbito de tarefas de classificação binária como nos caso de aproximação de funções. No primeiro caso, destacam-se as bases de dados de *benchmarking* de baixa e alta volumetria, bem como, as bases sintéticas utilizadas na avaliação qualitativa das fronteiras de decisão geradas pela propostas. No cenário de regressão, apresentam-se as bases de dados de *benchmarking* e se detalha o estudo de caso para a base de FOLSOM em que as duas propostas são avaliadas em termos de previsão de séries temporais de irradiância solar de curto prazo.

Destacou-se suas variáveis de entrada e saída, método de aquisição dos dados, métodos para comparação de desempenho e metodologia para pré-processamento da base. No próximo capítulo, a validação experimental de tais achados teóricos é realizada por meio de extensivas simulações numéricas sobre bases de dados reais e sintéticas, considerando tanto problemas de classificação como de regressão e cenários de baixa e alta volumetria. Ainda foi possível verificar a qualidade das fronteiras de decisão geradas em problemas de classificação binária sobre bases sintéticas e também foi avaliado o desempenho do método SCG-LSSVM

Figura 15 – *Pipeline* para a seleção de atributos.



Fonte: Elaborada pelo autor.

quando aplicado em uma base real de previsão de curto prazo de irradiância solar.

6 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os resultados obtidos para várias simulações computacionais que foram realizadas sobre base de dados reais e sintéticas a fim de validar as duas novas propostas. As propostas foram avaliadas para problemas de classificação binária e de aproximação de funções. Para um melhor detalhamento e organização, os resultados são apresentados considerando estas duas tarefas separadamente e, assim, as simulações para classificação são apresentadas na seção 6.1, enquanto os resultados para aproximação de funções são apresentados na seção 6.2.

6.1 Simulações para Classificação de Padrões

Os resultados obtidos para a tarefa de classificação binária, são apresentados e discutidos, levantando os principais ganhos e limitações das novas propostas, além de, trazer alguns delineamentos sobre possíveis melhorias que poderão ser implementadas em estudos futuros.

6.1.1 Base de Dados Pequenas

O primeiro ponto de avaliação dos modelos está na análise de seu desempenho preditivo quando considerado bases de dados pequenas. Inicia-se com a avaliação dos resultados de ajuste de hiperparâmetros realizados via otimização Bayesiana.

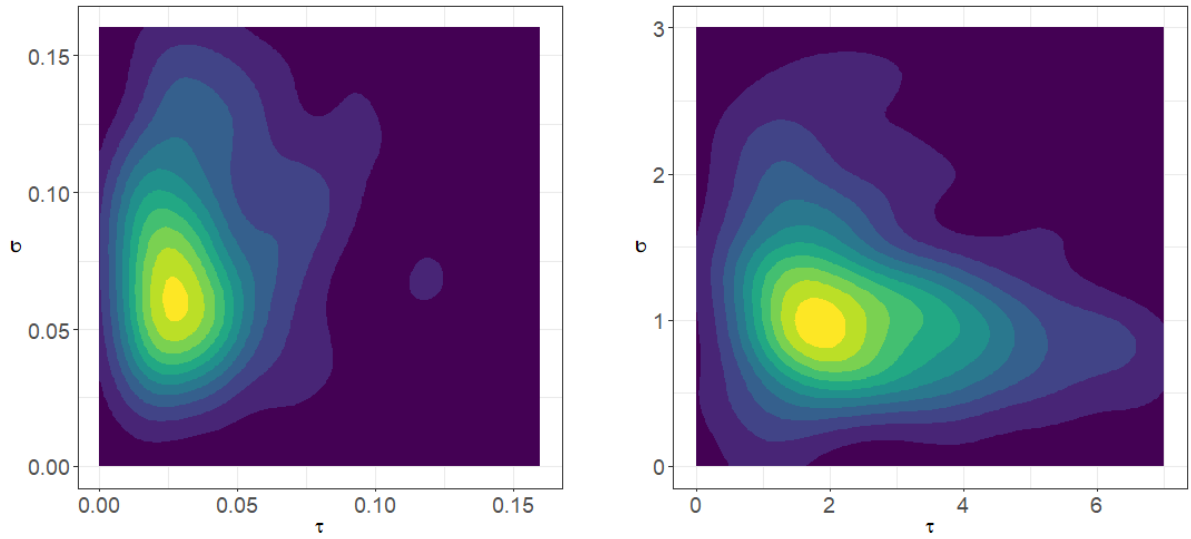
6.1.1.1 Ajuste de Hiperparâmetros

As Figuras 16, 17, 18 e 19 apresentam os resultados obtidos no processo de ajuste de hiperparâmetros para cada base de dados considerando o modelo TCSMO-LSSVM. Os resultados do ajuste de hiperparâmetros para o modelo SCG-LSSVM são apresentados nas Figuras 20, 21, 22 e 23. Além disso, a Tabela 11 apresenta os valores dos hiperparâmetros ótimos obtidos durante o processo de ajuste.

6.1.1.2 Desempenho Preditivo

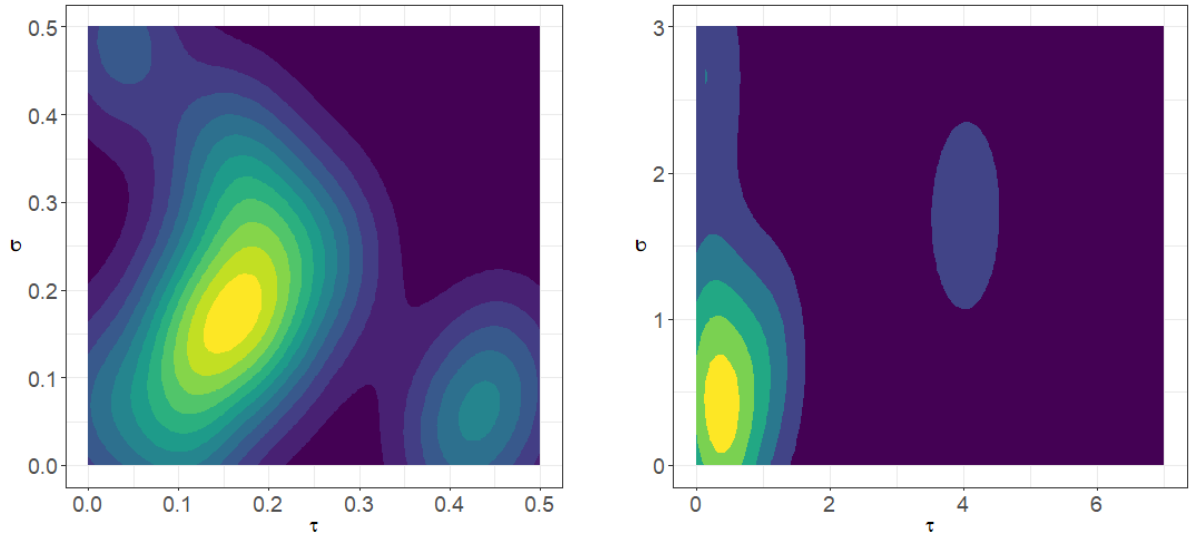
A Tabela 12 apresenta os resultados obtidos pelos classificadores LSSVM, P-LSSVM, IP-LSSVM, FSLM-LSSVM, CSMO-LSSVM e das duas propostas, TCSMO-LSSVM e SCG-

Figura 16 – Resultados para o ajuste de hiperparâmetros para as bases AUS (esquerda) e BCW (direita).



Fonte: Elaborada pelo autor.

Figura 17 – Resultados para o ajuste de hiperparâmetros para as bases BLD (esquerda) e VCP (direita).

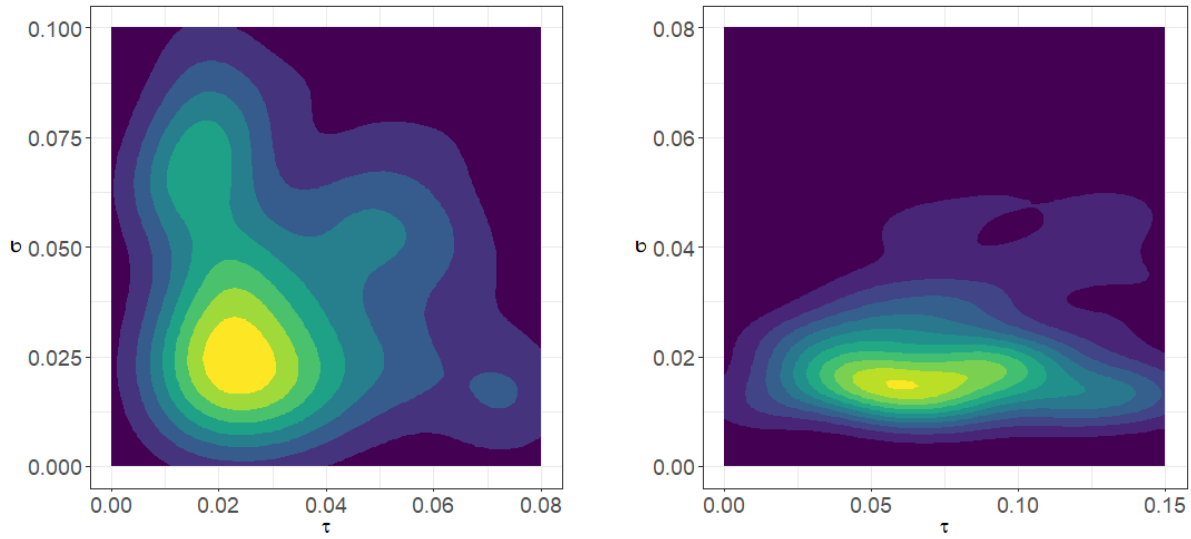


Fonte: Elaborada pelo autor.

LSSVM, quando aplicados a oito conjuntos de dados de *benchmarking* reais. Para obter o desempenho dos classificadores, foram realizadas 20 realizações independentes, nas quais calculou-se a média e o desvio padrão para a taxa de acerto das realizações. Também foi fornecida a porcentagem de redução no número de vetores-suporte em relação à quantidade do conjunto de vetores de treinamento. A porcentagem de poda selecionada para o TCSMO-LSSVM foi de 20%, pois esse valor permitiu o melhor equilíbrio entre esparsidade e capacidade preditiva.

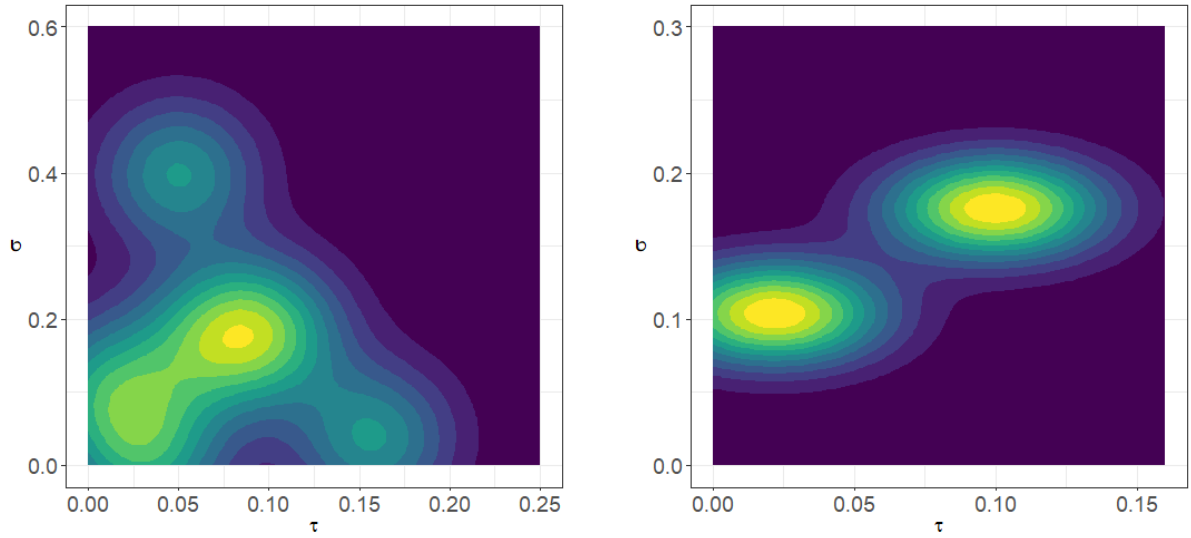
Esta decisão expõe um dilema entre ter uma maior nível de esparsidade da solução ótima do LSSVM com menor acurácia ou ter menor esparsidade com maior desempenho

Figura 18 – Resultados para o ajuste de hiperparâmetros para as bases PID (esquerda) e GCR (direita).



Fonte: Elaborada pelo autor.

Figura 19 – Resultados para o ajuste de hiperparâmetros para as bases HAB (esquerda) e ION (direita).

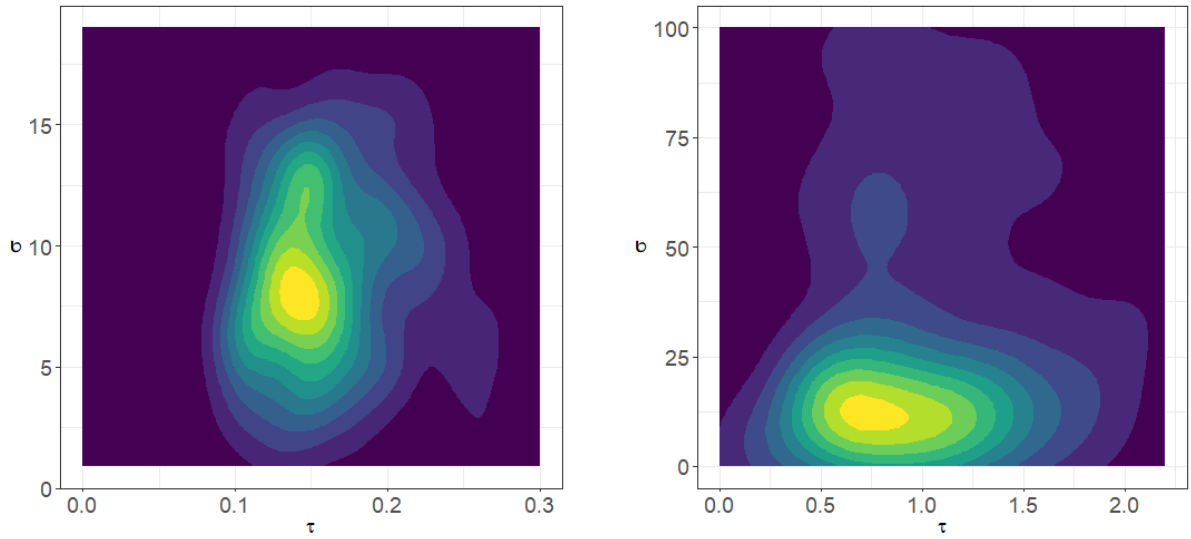


Fonte: Elaborada pelo autor.

preditivo. No caso, técnicas de reamostragem podem ser aplicadas com o intuito de obter uma estimativa ótima para o percentual de poda no TCSMO-LSSVM. Neste trabalho, definimos um *grid* de valores para a porcentagem de poda indo de 0% até 30% em passos de 5%, ou seja, foram considerados os valores de percentual de poda de $[0\%, 5\%, 10\%, 15\%, 20\%, 25\%, 30\%]$ para o TCSMO e avaliou-se a acurácia, bem como, o tempo de processamento do estágio de predição. Os resultados são reportados nas Figuras 24 e 25.

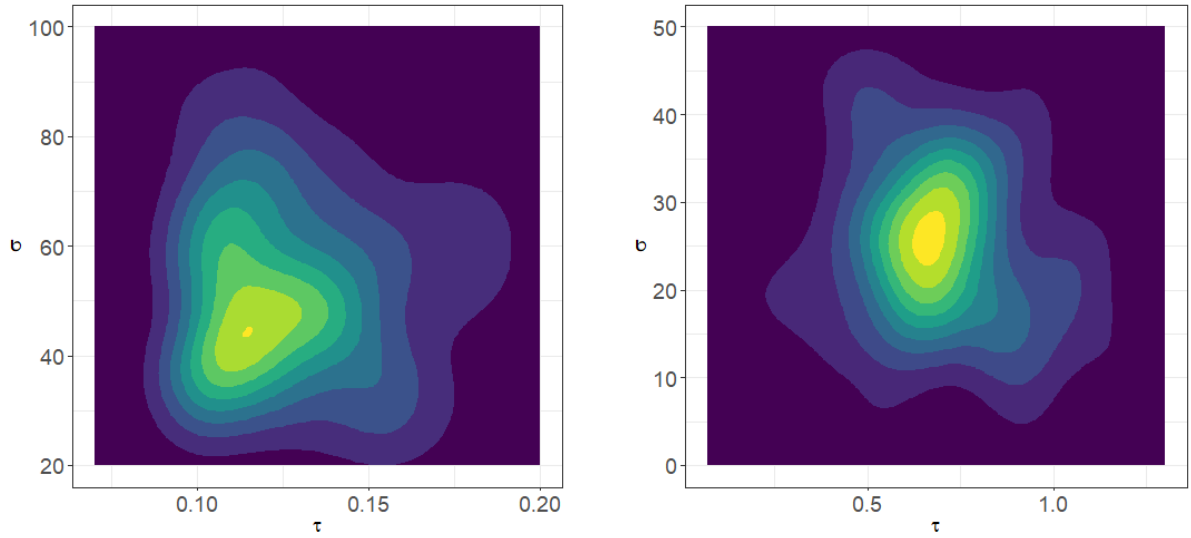
Os resultados apresentados mostram a porcentagem de redução dada pelo cômputo da expressão $1 - \frac{\#SV}{\#N}$, com a cardinalidade do conjunto de vetores de suporte dada por $\#SV$ e $\#N$

Figura 20 – Resultados para o ajuste de hiperparâmetros para as bases AUS (esquerda) e BCW (direita).



Fonte: Elaborada pelo autor.

Figura 21 – Resultados para o ajuste de hiperparâmetros para as bases BLD (esquerda) e VCP (direita).



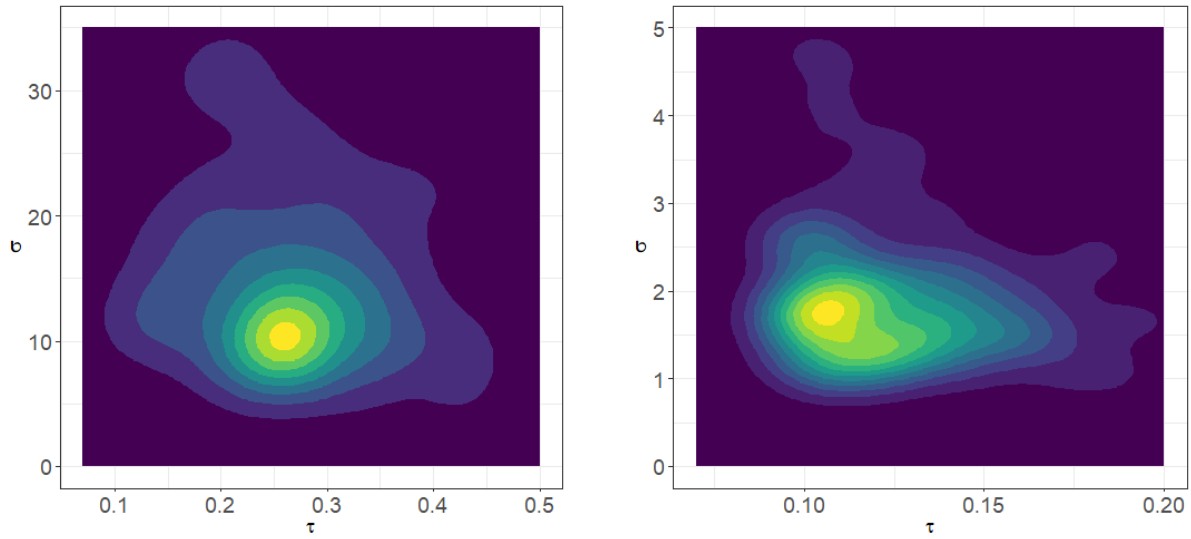
Fonte: Elaborada pelo autor.

o número de amostras de treinamento.

Um resumo dos resultados reportados na Tabela 12 é apresentado pelas Figuras 26 e 27. Com estes resultados é possível notar que a proposta TCSMO-LSSVM com percentual de poda de 20% apresenta desempenho competitivo quando comparado aos demais modelos considerados nas bases BCW, VCP, GCR e AUS. Nos demais conjuntos de dados, percebe-se um desempenho significativamente inferior ao LSSVM padrão.

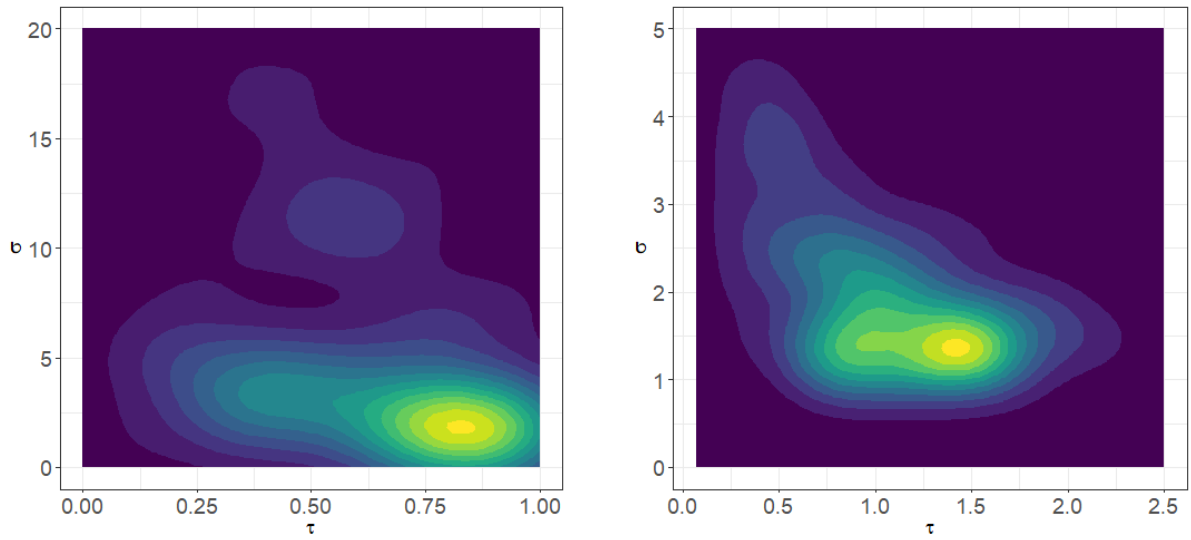
Este fato pode ser justificado pela convergência ruim do processo de ajuste do TCSMO-LSSVM para as bases de dados PID, BLD, ION e HAB, como fica evidente ao

Figura 22 – Resultados para o ajuste de hiperparâmetros para as bases PID (esquerda) e GCR (direita).



Fonte: Elaborada pelo autor.

Figura 23 – Resultados para o ajuste de hiperparâmetros para as bases HAB (esquerda) e ION (direita).



Fonte: Elaborada pelo autor.

analisar as Figuras 17, 18 e 19 em que claramente nota-se a ocorrência de múltiplos pontos de maximização da acurácia para cada par de hiperparâmetros.

Tais resultados podem ser consequência de uma exploração incompleta do espaço de busca, que pode ser contingenciada por um aumento no número de experimentos independentes na etapa de otimização dos parâmetros livres para estas bases. Com este refinamento, acredita-se que problemas de aprisionamento de ótimos locais possa ser superado, permitindo assim melhor desempenho para estas bases de dados.

Relativo a capacidade preditiva do modelo com relação ao percentual de poda,

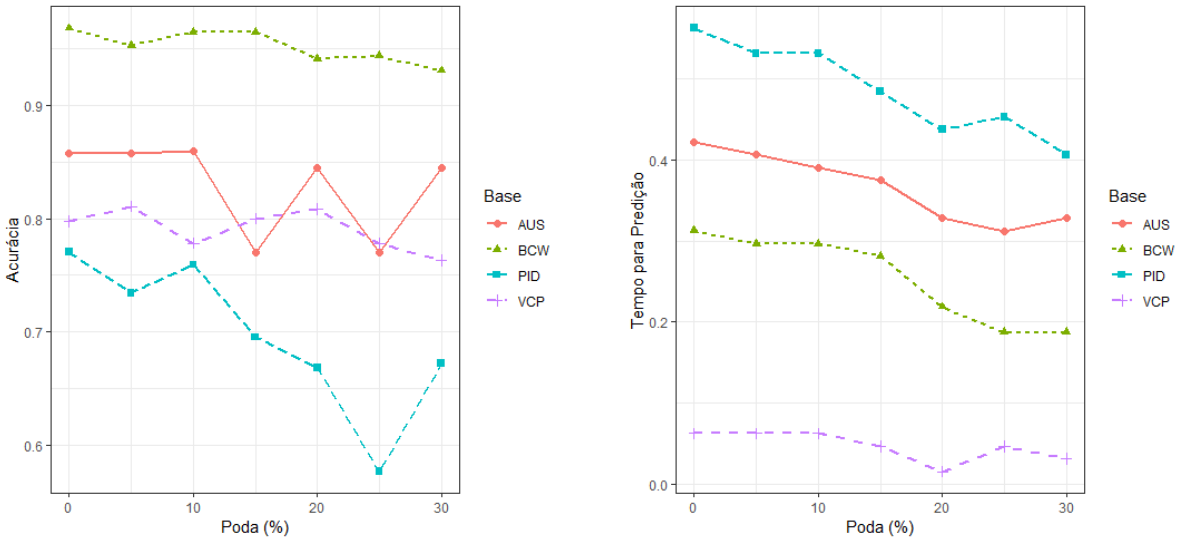
Tabela 11 – Valores ótimos para cada hiperparâmetro.

Base	Modelo	Hiperparâmetro	Valores
BCW	LSSVM	$[\sigma, \tau]$	$[1.47e^{-1}, 2.41e^{0}]$
BCW	P-LSSVM	$[\sigma, \tau]$	$[6.78e^{-2}, 2.40e^{0}]$
BCW	IP-LSSVM	$[\sigma, \tau]$	$[1.10e^{-1}, 6.18e^{0}]$
BCW	FSLM-LSSVM	$[\sigma, \tau]$	$[7.55e^{-1}, 4.36e^{0}]$
BCW	CSMO-LSSVM	$[\sigma, \tau]$	$[1.18e^{-1}, 1.26e^{0}]$
BCW	TCSMO-LSSVM	$[\sigma, \tau]$	$[1.48e^{-1}, 1.15e^{0}]$
BCW	SCG-LSSVM	$[\sigma, \tau]$	$[1.31e^{0}, 5.72e^{1}]$
VCP	LSSVM	$[\sigma, \tau]$	$[3.89e^{-1}, 1.23e^{0}]$
VCP	P-LSSVM	$[\sigma, \tau]$	$[7.17e^{-1}, 8.59e^{1}]$
VCP	IP-LSSVM	$[\sigma, \tau]$	$[9.32e^{-1}, 2.53e^{0}]$
VCP	FSLM-LSSVM	$[\sigma, \tau]$	$[9.41e^{-1}, 7.66e^{0}]$
VCP	CSMO-LSSVM	$[\sigma, \tau]$	$[8.46e^{-1}, 2.53e^{0}]$
VCP	TCSMO-LSSVM	$[\sigma, \tau]$	$[7.08e^{-1}, 7.80e^{0}]$
VCP	SCG-LSSVM	$[\sigma, \tau]$	$[6.65e^{-1}, 1.78e^{1}]$
PID	LSSVM	$[\sigma, \tau]$	$[1.18e^{-1}, 1.14e^{0}]$
PID	P-LSSVM	$[\sigma, \tau]$	$[3.93e^{-1}, 1.75e^{1}]$
PID	IP-LSSVM	$[\sigma, \tau]$	$[2.14e^{-1}, 2.80e^{0}]$
PID	FSLM-LSSVM	$[\sigma, \tau]$	$[7.45e^{-1}, 3.30e^{0}]$
PID	CSMO-LSSVM	$[\sigma, \tau]$	$[1.03e^{-1}, 3.80e^{0}]$
PID	TCSMO-LSSVM	$[\sigma, \tau]$	$[2.52e^{-1}, 2.24e^{0}]$
PID	SCG-LSSVM	$[\sigma, \tau]$	$[2.63e^{-1}, 1.86e^{1}]$
GCR	LSSVM	$[\sigma, \tau]$	$[1.02e^{-1}, 1.29e^{0}]$
GCR	P-LSSVM	$[\sigma, \tau]$	$[9.22e^{0}, 8.95e^{-2}]$
GCR	IP-LSSVM	$[\sigma, \tau]$	$[1.91e^{-1}, 4.59e^{0}]$
GCR	FSLM-LSSVM	$[\sigma, \tau]$	$[2.35e^{-1}, 7.24e^{0}]$
GCR	CSMO-LSSVM	$[\sigma, \tau]$	$[1.30e^{-1}, 1.78e^{0}]$
GCR	TCSMO-LSSVM	$[\sigma, \tau]$	$[1.28e^{-1}, 2.72e^{0}]$
GCR	SCG-LSSVM	$[\sigma, \tau]$	$[1.01e^{-1}, 1.79e^{0}]$
HAB	LSSVM	$[\sigma, \tau]$	$[1.60e^{-1}, 1.29e^{0}]$
HAB	P-LSSVM	$[\sigma, \tau]$	$[6.73e^{-1}, 8.47e^{0}]$
HAB	IP-LSSVM	$[\sigma, \tau]$	$[8.36e^{-1}, 1.04e^{0}]$
HAB	FSLM-LSSVM	$[\sigma, \tau]$	$[1.69e^{-1}, 5.92e^{0}]$
HAB	CSMO-LSSVM	$[\sigma, \tau]$	$[3.36e^{-1}, 1.04e^{0}]$
HAB	TCSMO-LSSVM	$[\sigma, \tau]$	$[2.73e^{-1}, 4.62e^{0}]$
HAB	SCG-LSSVM	$[\sigma, \tau]$	$[3.95e^{-2}, 2.31e^{0}]$
BLD	LSSVM	$[\sigma, \tau]$	$[1.27e^{-1}, 1.41e^{0}]$
BLD	P-LSSVM	$[\sigma, \tau]$	$[1.23e^{0}, 9.86e^{-1}]$
BLD	IP-LSSVM	$[\sigma, \tau]$	$[2.79e^{-1}, 2.16e^{0}]$
BLD	FSLM-LSSVM	$[\sigma, \tau]$	$[2.00e^{-1}, 1.10e^{0}]$
BLD	CSMO-LSSVM	$[\sigma, \tau]$	$[1.40e^{-1}, 4.17e^{0}]$
BLD	TCSMO-LSSVM	$[\sigma, \tau]$	$[2.57e^{-1}, 3.81e^{0}]$
BLD	SCG-LSSVM	$[\sigma, \tau]$	$[1.01e^{-1}, 3.95e^{1}]$
ION	LSSVM	$[\sigma, \tau]$	$[1.16e^{-1}, 1.18e^{0}]$
ION	P-LSSVM	$[\sigma, \tau]$	$[1.43e^{-1}, 1.72e^{1}]$
ION	IP-LSSVM	$[\sigma, \tau]$	$[2.48e^{-1}, 1.30e^{0}]$
ION	FSLM-LSSVM	$[\sigma, \tau]$	$[5.23e^{-1}, 1.96e^{0}]$
ION	CSMO-LSSVM	$[\sigma, \tau]$	$[8.70e^{-1}, 7.78e^{0}]$
ION	TCSMO-LSSVM	$[\sigma, \tau]$	$[1.90e^{-1}, 1.27e^{0}]$
ION	SCG-LSSVM	$[\sigma, \tau]$	$[2.06e^{-1}, 2.77e^{1}]$
AUS	LSSVM	$[\sigma, \tau]$	$[1.11e^{-1}, 2.73e^{0}]$
AUS	P-LSSVM	$[\sigma, \tau]$	$[3.90e^{-1}, 3.05e^{0}]$
AUS	IP-LSSVM	$[\sigma, \tau]$	$[1.85e^{-1}, 1.85e^{0}]$
AUS	FSLM-LSSVM	$[\sigma, \tau]$	$[2.38e^{-1}, 3.52e^{0}]$
AUS	CSMO-LSSVM	$[\sigma, \tau]$	$[3.07e^{-1}, 6.54e^{0}]$
AUS	TCSMO-LSSVM	$[\sigma, \tau]$	$[3.72e^{-1}, 1.72e^{0}]$
AUS	SCG-LSSVM	$[\sigma, \tau]$	$[1.49e^{-1}, 3.47e^{0}]$

Fonte: Elaborada pelo autor.

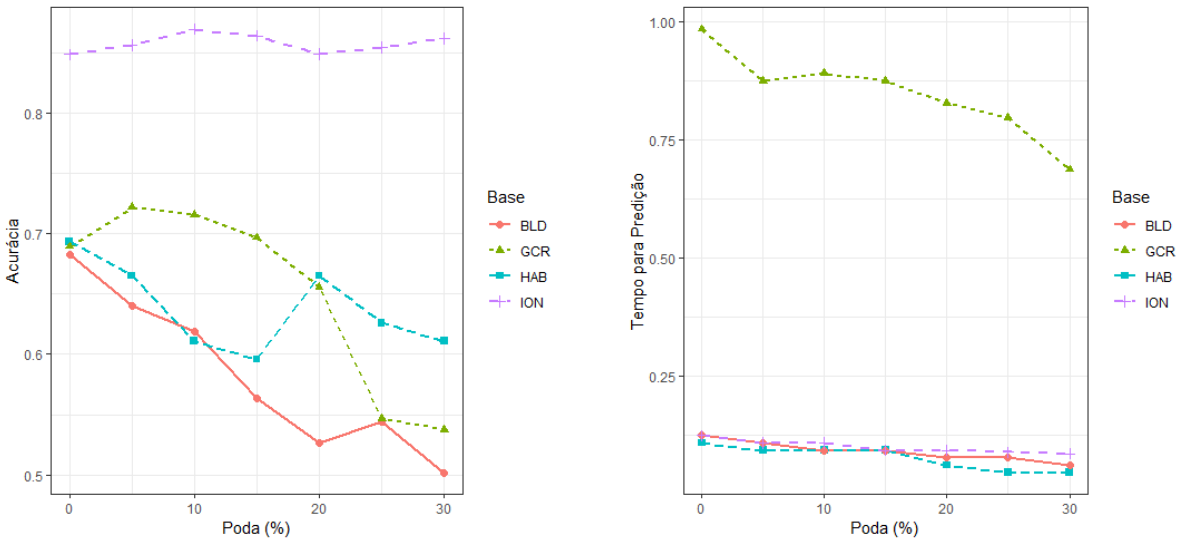
pode-se perceber pela análise das Figuras 24 e 25 o padrão geral de queda com o aumento do percentual, o que é convergente com os resultados reportados em Dias e Neto (2017), Oliveira *et al.* (2018), Dias *et al.* (2018), vale destacar que para algumas bases como para AUS, HAB e PID houve ampliação ou oscilação da acurácia com o aumento do percentual, o que pode indicar a necessidade de implementações de melhorias na metodologia de poda adotada para

Figura 24 – Acurácia e tempo de previsão em termos de porcentagem de poda no TCSMO-LSSVM para conjuntos de dados AUS, BCW, PID e VCP.



Fonte: Elaborada pelo autor.

Figura 25 – Acurácia e tempo de previsão em termos de porcentagem de poda no TCSMO-LSSVM para conjuntos de dados BLD, GCR, HAB e ION.



Fonte: Elaborada pelo autor.

o TCSMO-LSSVM, uma vez que, tais oscilações podem indicar a poda de padrões que são relevantes para a construção da fronteira de decisão. Além disso, o tempo no estágio de predição tem comportamento monotônico decrescente relativo ao percentual de poda, o que faz sentido, já que com o aumento do percentual de poda, menor o número de padrões de treinamento utilizados na predição de um novo registro (Zeng e Chen, 2005).

Ainda é possível notar o rápido treinamento das abordagens baseadas no problema dual do LSSVM utilizando variantes do algoritmo SMO. De fato, muitos trabalhos já constataram o rápido treinamento de modelos SVM e LSSVM por meio do emprego do algoritmo SMO e

Tabela 12 – Resultados para classificadores treinados de acordo com a metodologia 70/30 usando *kernel* Gaussiano Radial.

Base	**Classificador**	**Acurácia**(%)	**Redução** (%)	**Tempo**(s)
BCW	LSSVM	93.57 ± 3.43	0	$3.61e^{3} \pm 11.63$
BCW	P-LSSVM	86.67 ± 2.60	37.35	$2.76e^{2} \pm 30.13$
BCW	IP-LSSVM	89.82 ± 2.36	10.83	$8.72e^{1} \pm 11.65$
BCW	FSLM-LSSVM	95.56 ± 0.67	20	$5.82e0 \pm 0.54$
BCW	CSMO-LSSVM	96.60 ± 1.62	0	$1.21e^{0} \pm 0.15$
BCW	TCSMO-LSSVM	93.68 ± 2.63	20	$1.49e^{0} \pm 0.06$
BCW	SCG-LSSVM	90.41 ± 2.0	79.78	$2.96e^{0} \pm 0.59$
VCP	LSSVM	83.87 ± 2.40	0	$6.01e2 \pm 9.04$
VCP	P-LSSVM	75.48 ± 9.36	50.33	$6.50e^{1} \pm 2.12$
VCP	IP-LSSVM	76.13 ± 3.98	19.87	$2.41e^{1} \pm 5.55$
VCP	FSLM-LSSVM	79.35 ± 3.08	20	$4.45e^{0} \pm 0.16$
VCP	CSMO-LSSVM	78.92 ± 2.10	0	$3.50e^{-1} \pm 0.02$
VCP	TCSMO-LSSVM	79.78 ± 5.95	20	$4.72e^{-1} \pm 0.11$
VCP	SCG-LSSVM	78.49 ± 5.32	78.80	$2.30e^{0} \pm 0.96$
PID	LSSVM	77.92 ± 3.03	0	$1.01e^{4} \pm 27.49$
PID	P-LSSVM	72.55 ± 4.39	43.47	$5.34e^{1} \pm 9.34$
PID	IP-LSSVM	71.08 ± 4.54	8	$1.39e^{2} \pm 1.38$
PID	FSLM-LSSVM	73.68 ± 1.6	20	$1.08e^{1} \pm 0.11$
PID	CSMO-LSSVM	66.93 ± 9.10	0	$1.93e^{0} \pm 0.03$
PID	TCSMO-LSSVM	69.52 ± 7.39	20	$2.28e^{0} \pm 0.12$
PID	SCG-LSSVM	72.38 ± 2.15	79.73	$5.23e^{0} \pm 1.75$
GCR	LSSVM	74.80 ± 1.73	0	$9.24e^{3} \pm 10.73$
GCR	P-LSSVM	71.53 ± 3.30	32.11	$1.14e^{2} \pm 1.96$
GCR	IP-LSSVM	72.07 ± 3.70	6.13	$2.44e^{2} \pm 2.57$
GCR	FSLM-LSSVM	74.93 ± 3.58	20	$1.27^{1} \pm 0.13$
GCR	CSMO-LSSVM	69.47 ± 2.54	0	$3.64e^{0} \pm 0.05$
GCR	TCSMO-LSSVM	70.20 ± 1.17	20	$3.77e^{0} \pm 0.09$
GCR	SCG-LSSVM	69.67 ± 2.37	79.96	$8.62e^{0} \pm 3.23$
HAB	LSSVM	73.04 ± 5.29	0	$1.57e^{3} \pm 30.76$
HAB	P-LSSVM	73.91 ± 3.99	36.26	$3.34e^{1} \pm 2.72$
HAB	IP-LSSVM	71.74 ± 2.77	17.54	$2.23e^{1} \pm 6.08$
HAB	FSLM-LSSVM	71.74 ± 2.77	20	$4.26e^{0} \pm 0.15$
HAB	CSMO-LSSVM	69.13 ± 5.73	0	$3.28e^{-1} \pm 0.03$
HAB	TCSMO-LSSVM	64.78 ± 5.35	20	$5.88e^{-1} \pm 0.03$
HAB	SCG-LSSVM	73.91 ± 3.61	67.11	$1.39e^{1} \pm 0.32$
BLD	LSSVM	67.88 ± 5.33	0	$1.12e^{4} \pm 46.1$
BLD	P-LSSVM	63.27 ± 4.10	36.26	$5.64e^{1} \pm 3.17$
BLD	IP-LSSVM	58.85 ± 5.34	17.54	$2.89e^{1} \pm 7.55$
BLD	FSLM-LSSVM	59.61 ± 3.26	20	$4.24e^{0} \pm 0.09$
BLD	CSMO-LSSVM	62.88 ± 1.99	0	$4.22e^{-1} \pm 0.05$
BLD	TCSMO-LSSVM	54.23 ± 8.34	20	$6.69e^{-1} \pm 0.15$
BLD	SCG-LSSVM	56.73 ± 5.93	77.98	$4.03e^{0} \pm 1.81$
ION	LSSVM	93.01 ± 2.18	0	$4.83e^{2} \pm 16.81$
ION	P-LSSVM	89.81 ± 2.44	36.26	$2.39e^{1} \pm 8.08$
ION	IP-LSSVM	91.70 ± 2.78	17.54	$2.94e^{1} \pm 1.68$
ION	FSLM-LSSVM	86.79 ± 2.31	20	$3.73e^{0} \pm 0.20$
ION	CSMO-LSSVM	87.36 ± 1.07	0	$4.25e^{-1} \pm 0.01$
ION	TCSMO-LSSVM	82.26 ± 1.39	20	$7.84e^{-1} \pm 0.11$
ION	SCG-LSSVM	80.38 ± 8.83	22.81	$2.37e^{0} \pm 1.23$
AUS	LSSVM	84.35 ± 2.70	0	$4.78e^{3} \pm 13.88$
AUS	P-LSSVM	82.22 ± 3.81	36.26	$7.73e^{1} \pm 1.67$
AUS	IP-LSSVM	85.41 ± 2.87	17.54	$1.21e^{2} \pm 2.79$
AUS	FSLM-LSSVM	72.27 ± 3.21	20	$7.97e^{0} \pm 0.24$
AUS	CSMO-LSSVM	81.26 ± 4.80	0	$1.68e^{0} \pm 0.06$
AUS	TCSMO-LSSVM	85.31 ± 3.72	20	$1.95e^{0} \pm 0.10$
AUS	SCG-LSSVM	80.97 ± 4.29	79.82	$2.12e^{0} \pm 0.16$

Fonte: Elaborada pelo autor.

muito se deve a decomposição do problema dual em subproblemas, graças ao uso da estratégia de selecionar pares de multiplicadores de Lagrange para a atualização a cada iteração. Avaliar apenas dois multiplicadores por vez, garante uma solução analítica fechada para o subproblema permitindo uma grande simplificação da computação da atualização das variáveis duais.

Para todas as bases também constatou-se que o tempo necessário para o treinamento no modelo TCSMO-LSSVM é praticamente equivalente ao CSMO-LSSVM. Este é um resultado interessante, pois o número de flops necessário a cada iteração do TCSMO-LSSVM é superior ao CSMO-LSSVM, no entanto, como já foi discutido na Seção 4, o ganho funcional desenvolvido pelo uso do TCSMO é superior ao do CSMO, bem como, ao do SMO de primeira e segunda ordem, portanto, por mais que o custo computacional por iteração do TCSMO-LSSVM seja superior ao CSMO-LSSVM, seu ganho funcional equilibra esse maior custo na atualização da variável dual, o que justifica essa similaridade nos tempos de treinamento dos modelos TCSMO-LSSVM e CSMO-LSSVM (Yu *et al.*, 2023a).

Relativo a segunda proposta, SCG-LSSVM, nota-se que seu desempenho em termos de acurácia é um pouco inferior ao TCSMO-LSSVM no geral. Percebe-se problemas de convergência para este método nas bases BLD, GCR e ION, onde obteve acurácias significativamente inferiores ao LSSVM padrão, mas similar às demais variantes esparsas. Novamente, estes resultados podem ser justificados pela ocorrência de problemas relacionados a obtenção de ótimos locais para o desempenho dos modelos. Portanto, assim como no caso da primeira proposta, a ampliação do espaço de busca, bem como, o aumento do número de *trials* dentro do processo de otimização Bayesiana seriam soluções possíveis.

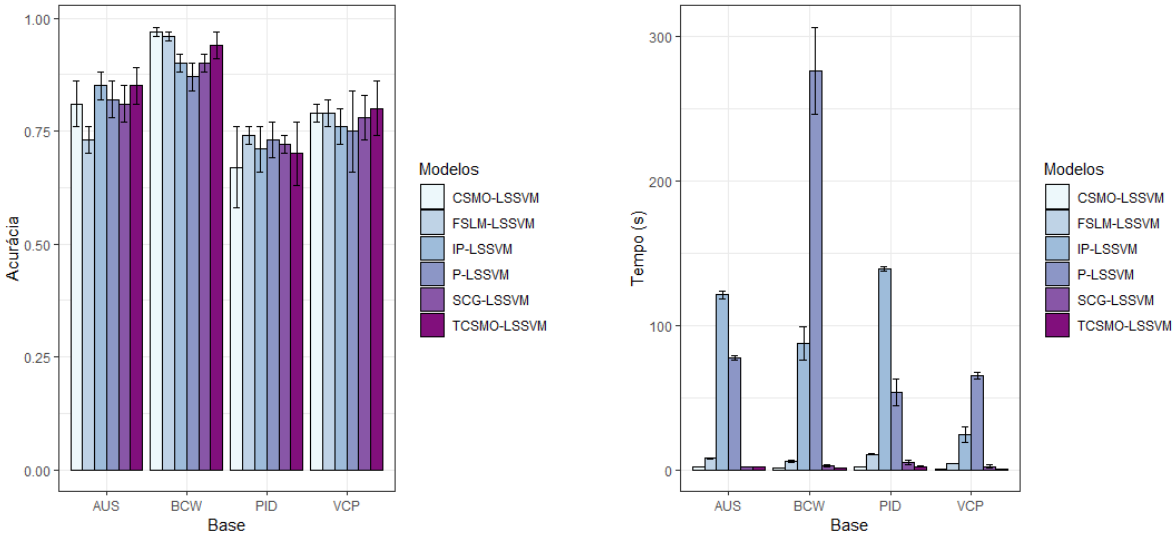
Destaca-se o rápido treinamento do SCG-LSSVM sendo bem mais eficiente do que as propostas que se baseiam na solução do problema primal para o treinamento do modelo LSSVM. Isto é um indicativo de que o emprego de uma abordagem dual pode ser mais eficiente para o treinamento do modelo LSSVM do que a abordagem primal, já que os todos os modelos que utilizam as abordagens duais, SCG-LSSVM, CSMO-LSSVM e TCSMO-LSSVM, apresentam treinamento mais rápidos do que as demais.

O rápido treinamento do modelo SCG-LSSVM também deve-se ao seu algoritmo de otimização que usa informações de segunda ordem da função objetivo, $\mathscr{D}(\boldsymbol{\alpha})$, no caso a Hessiana, o que permite um elevado ganho funcional por interação já que a aproximação para a função objetivo na sua vizinhança é melhor, a medida que se considera mais termos em sua aproximação de Taylor (Izmailov e Solodov, 2007).

Diferentemente dos métodos de Newton clássicos, que usam a Hessiana diretamente, nossa proposta utiliza apenas uma aproximação BFGS da mesma, que é atualizada a cada iteração o que indica que o algoritmo de otimização SCG-LSSVM é Quasi-Newton, similarmente ao algoritmo de Levenberg-Marquardt. Isto induz uma redução no custo de operações por iteração

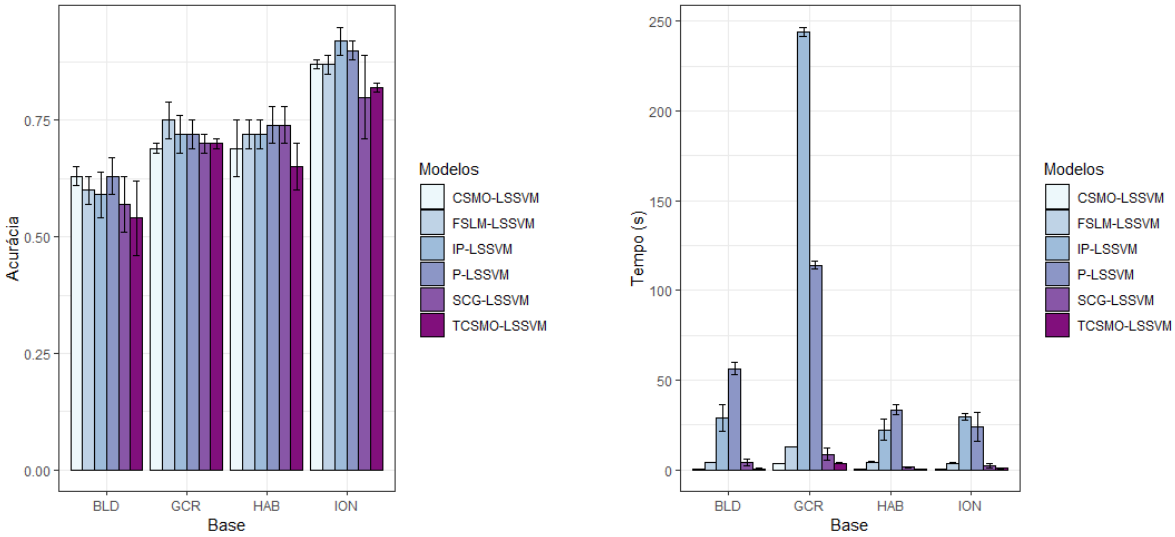
do método contribuindo ainda mais para a sua eficiência (Levenberg, 1944; Marquardt, 1963).

Figura 26 – Acurácia e tempo de treinamento para cada modelo nos conjuntos de dados AUS, BCW, PID e VCP.



Fonte: Elaborada pelo autor.

Figura 27 – Acurácia e tempo de treinamento para cada modelo nos conjuntos de dados BLD, GCR, HAB e ION.

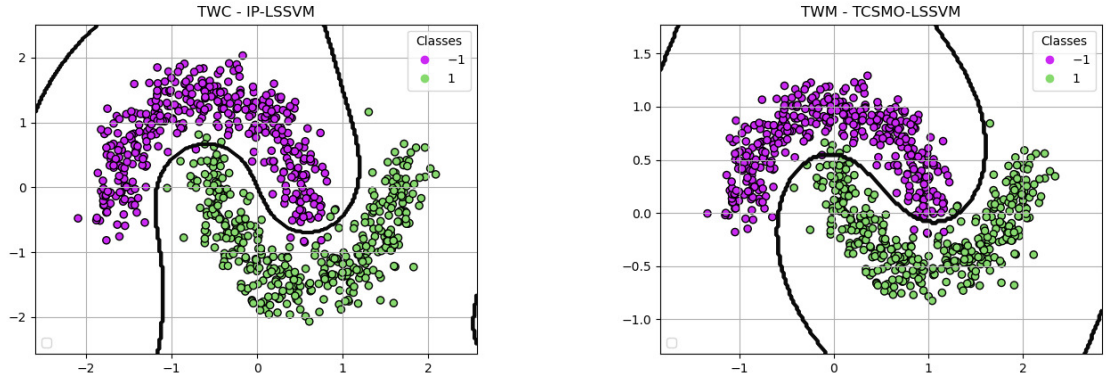


Fonte: Elaborada pelo autor.

6.1.2 Fronteira de Decisão

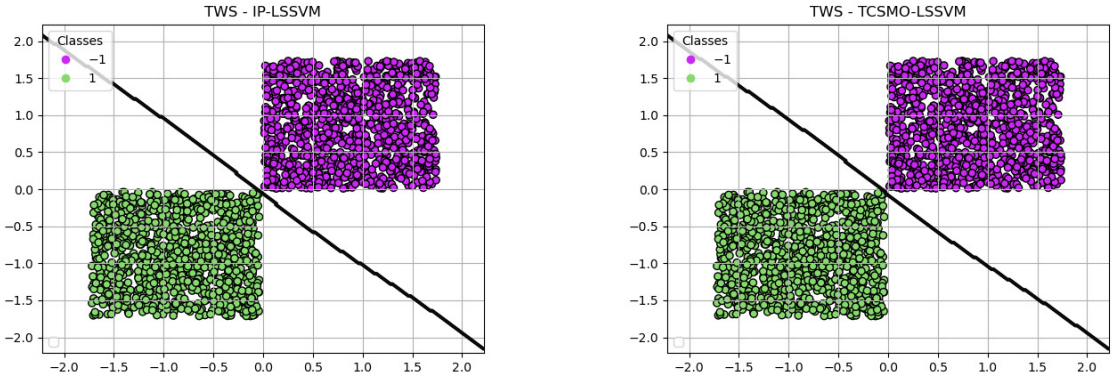
Pela observação das Figuras 28, 29 e 30, é possível notar que a fronteira de decisão gerada pelo TCSMO-LSSVM é bastante similar a fronteira desenvolvida pelo IP-LSSVM o que ratifica o bom ajuste do TCSMO-LSSVM aos dados sintéticos. Este fato indica que a poda

Figura 28 – Fronteira de decisão gerada pelos modelos IP-LSSVM e TCSMO-LSSVM para a base TWM.



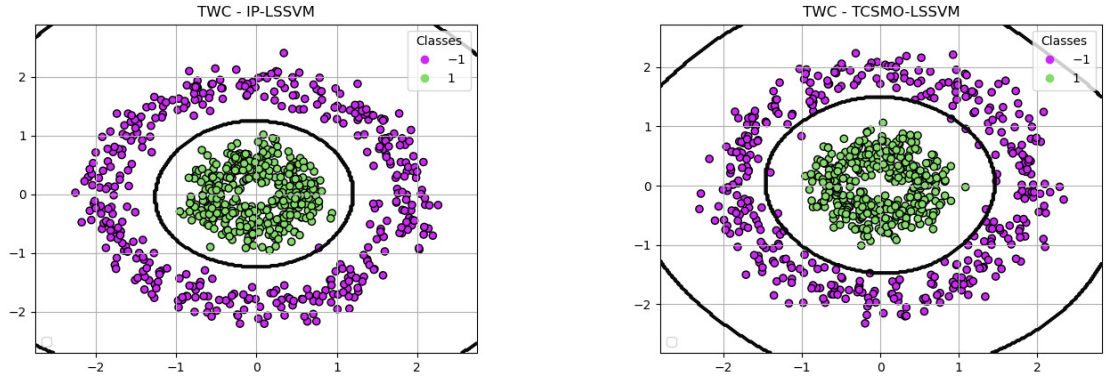
Fonte: Elaborada pelo autor.

Figura 29 – Fronteira de decisão gerada pelos modelos IP-LSSVM e TCSMO-LSSVM para a base TWS.



Fonte: Elaborada pelo autor.

Figura 30 – Fronteira de decisão gerada pelos modelos IP-LSSVM e TCSMO-LSSVM para a base TWC.



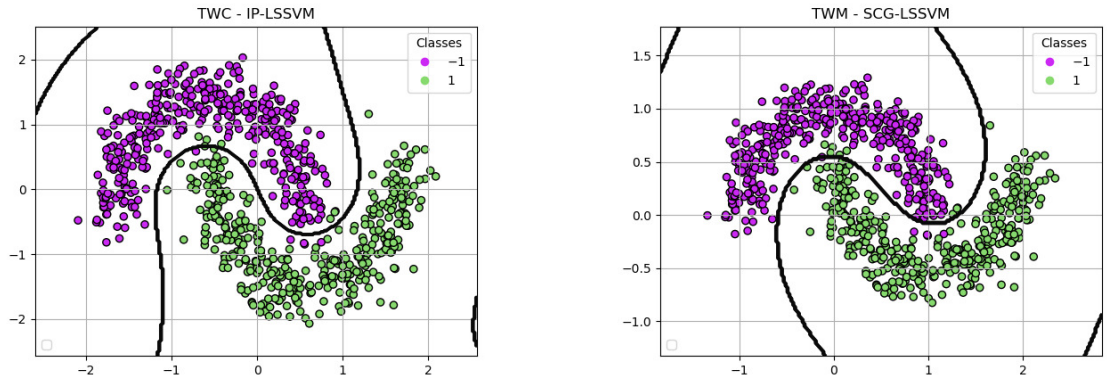
Fonte: Elaborada pelo autor.

através do critério de ganho funcional é adequada para estas bases e que os padrões menos representativos para a formação da fronteira estão sendo corretamente identificados e removidos.

Nota-se que as fronteiras de decisão obtidas pelo IP-LSSVM são mais sobreajustadas,

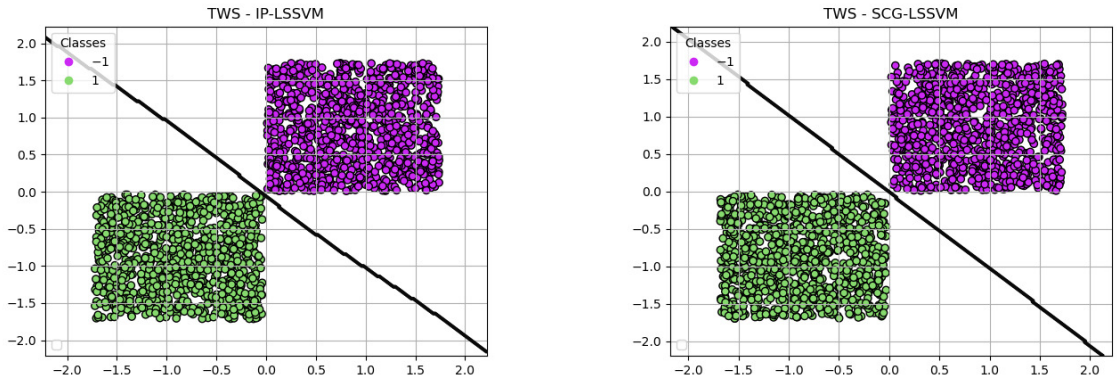
ou seja, penaliza padrões classificados erroneamente, como observado na Figura 30, que mostra que a fronteira de decisão interna está mais bem posicionada do que aquela obtida pelo modelo TCSMO-LSSVM. De uma forma geral, é relevante a obtenção de heurísticas que permitam a estimação de um percentual ótimo de poda seja por técnicas de reamostragem ou diretamente dentro da atualização dos parâmetros do modelo. Estas são necessárias e serão consideradas em estudos futuros.

Figura 31 – Fronteira de decisão gerada pelos modelos IP-LSSVM e SCG-LSSVM para a base TWM.



Fonte: Elaborada pelo autor.

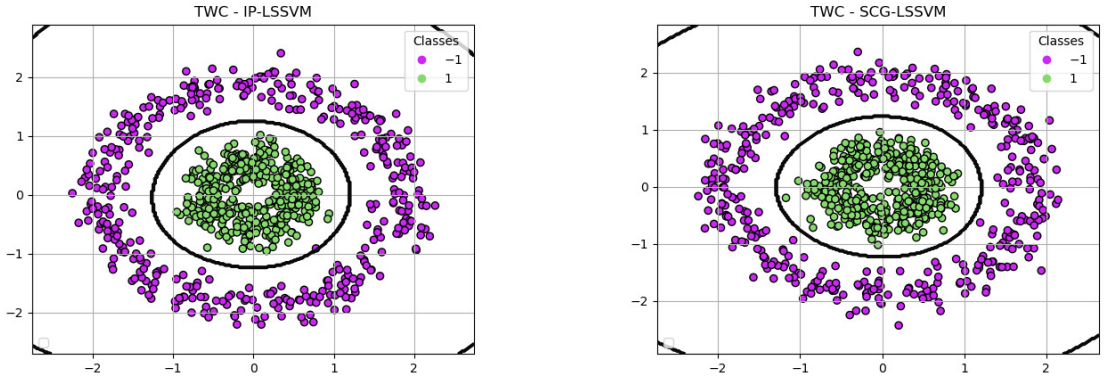
Figura 32 – Fronteira de decisão gerada pelos modelos IP-LSSVM e SCG-LSSVM para a base TWS.



Fonte: Elaborada pelo autor.

Pela observação das Figuras 31, 32 e 33, nota-se que a fronteira de decisão gerada pelo SCG-LSSVM consegue realizar a separação entre as duas classes para as bases sintéticas, mantendo grande similaridade com a fronteira desenvolvida pelo modelo IP-LSSVM. Destaca-se o bom posicionamento da fronteira em todos os casos, bem como seu formato tanto nos casos lineares como nos não lineares.

Figura 33 – Fronteira de decisão gerada pelos modelos IP-LSSVM e SCG-LSSVM para a base TWC.



Fonte: Elaborada pelo autor.

De uma forma geral, é possível determinar que a proposta SCG-LSSVM tem boa capacidade de discriminação das classes, bem como, gera uma fronteira de decisão menos flexível, ou seja, esta tenta seguir menos os padrões quando comparada a desenvolvida pelo IP-LSSVM o que pode ser visto como algo benéfico, uma vez que este comportamento tende a evitar problemas de sobreajuste.

6.1.3 Bases de Dados Grandes

Neste ponto, vale destacar que foram considerados apenas os modelos CSMO-LSSVM, TCSMO-LSSVM sem poda, TCSMO-LSSVM com 20% de poda e o SCG-LSSVM. Devido a limitações de processamento o que fez com que algumas experimentações computacionais demorassem tempo demais, ocasionando em algumas situações até mesmo a reinicialização da máquina, os demais modelos baseados no problema primal não foram considerados, este fato serve para ratificar o rápido treinamento dos modelos LSSVM baseados no problema dual, bem como, a grande eficiência de modelos baseados no algoritmo SMO (Platt, 1998; Yu *et al.*, 2023a; Yu *et al.*, 2023b). Os resultados obtidos são reportados na Tabela 13.

Pela avaliação dos resultados da Tabela 13, nota-se que o TCSMO-LSSVM sem poda conseguiu convergir para todos as bases analisadas, obtendo altos valores de acurácia, confirmando a capacidade que tal proposta tem de tratar com grandes bases de dados. Outra característica observada destes resultados é o fato de o CSMO-LSSVM ter apresentado baixas acurácias para as bases BKM e ADT. De fato, por mais que tal modelo consiga gerar resultados para estes dados em tempo hábil, o mesmo apresenta problemas de convergência para estes dois conjuntos, o que pode ser um indicativo de instabilidade do modelo relativo ao número

Tabela 13 – Acurácia e tempo de treinamento para as grandes bases de dados.

Base	Classificador	Acurácia	Tempo (s)
BKM	TCSMO-LSSVM	85 ± 1	$6.58e^1 \pm 0.32$
BKM	TCSMO-LSSVM (20%)	85 ± 3	$8.65e^1 \pm 1.22$
BKM	SCG-LSSVM	89.17 ± 0.66	$6.59e^1 \pm 0.25$
BKM	CSMO-LSSVM	58.54 ± 23.6	$6.44e^1 \pm 0.23$
ADT	TCSMO-LSSVM	85 ± 1	$1.83e^3 \pm 32.04$
ADT	TCSMO-LSSVM (20%)	76 ± 1	$1.69e^3 \pm 65.56$
ADT	SCG-LSSVM	87.92 ± 0.73	$1.81e^3 \pm 8.00$
ADT	CSMO-LSSVM	65.12 ± 11.96	$1.75e^3 \pm 11.69$
SHT	TCSMO-LSSVM	99.8 ± 1	$3.16e^3 \pm 16.14$
SHT	TCSMO-LSSVM (20%)	99.8 ± 1	$3.01e^3 \pm 58.39$
SHT	SCG-LSSVM	99.87 ± 0.1	$3.14e^3 \pm 38.87$
SHT	CSMO-LSSVM	98.14 ± 0.5	$2.91e^3 \pm 47.42$

Fonte: Elaborada pelo autor.

de entradas, vale frisar que a convergência nos dados SHT é facilitada pelo alto nível de desbalanceamento desta base, permitindo altas acurácias até mesmo para o CSMO-LSSVM.

Para o TCSMO-LSSVM com percentual de poda de 20%, percebe-se uma menor instabilidade da capacidade discriminativa do modelo com o aumento no número de padrões. De fato, para as bases BKM e SHT o desempenho entre o TCSMO-LSSVM com e sem poda é praticamente o mesmo. Para a base ADT, tem-se uma queda mais considerável da acurácia do TCSMO-LSSVM $(\%20)$ quando comparado ao TCSMO-LSSVM sem poda, no entanto, pela variância das acurácias avaliadas em cada rodada dentro da validação é possível notar que o modelo convergiu provavelmente para um ótimo local.

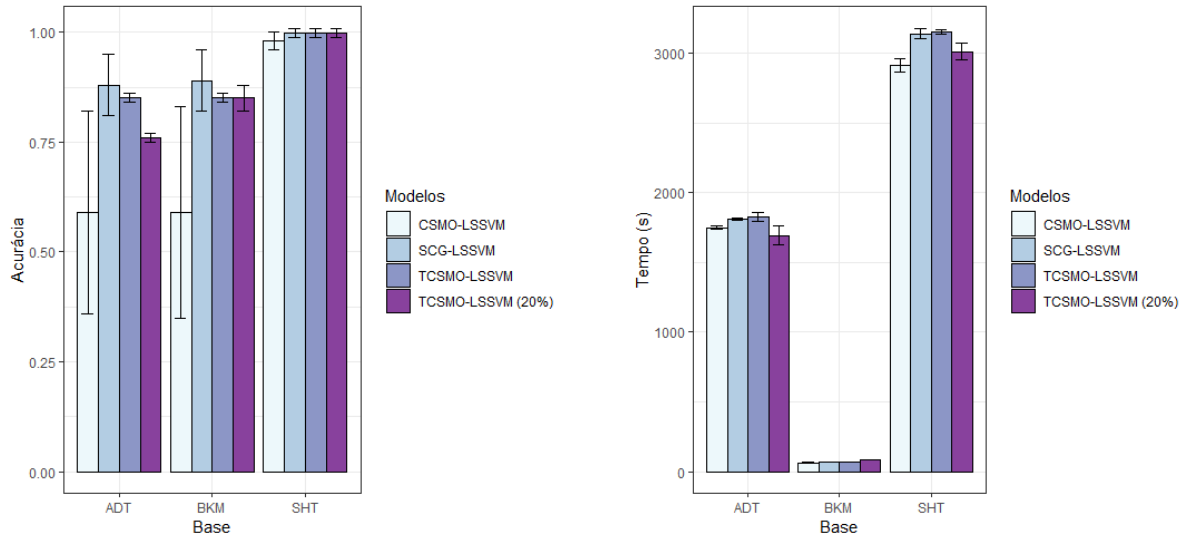
Para o caso do SCG-LSSVM, percebe-se convergência para todos os grandes conjuntos de dados analisados, alcançando altos valores de acurácia, confirmando a capacidade desta proposta de lidar com grandes bases de dados.

De uma forma geral, nota-se que as propostas TCSMO-LSSVM e SCG-LSSVM conseguem processar grandes bases em tempo hábil e com convergência do algoritmo de otimização, ratificando que estas novas variantes do LSSVM trazem os benefícios de robustez do LSSVM com o acréscimo de rápido treinamento, capacidade de tratar com grandes bases de dados e obtenção de soluções esparsas.

6.2 Simulações para Aproximações de Funções

O resultados para a avaliação das propostas na tarefa de aproximação de funções são apresentados e discutidos, destacando-se os ganhos e limitações obtidas, bem como, delineando

Figura 34 – Acurácia e tempo de treinamento para cada modelo nos conjuntos de dados ADT, BKM, SHT.



Fonte: Elaborada pelo autor.

possíveis melhorias a serem realizadas em estudos futuros, de uma forma similar ao que foi feito na seção de resultados para classificação binária.

6.2.1 Ajuste de Hiperparâmetros

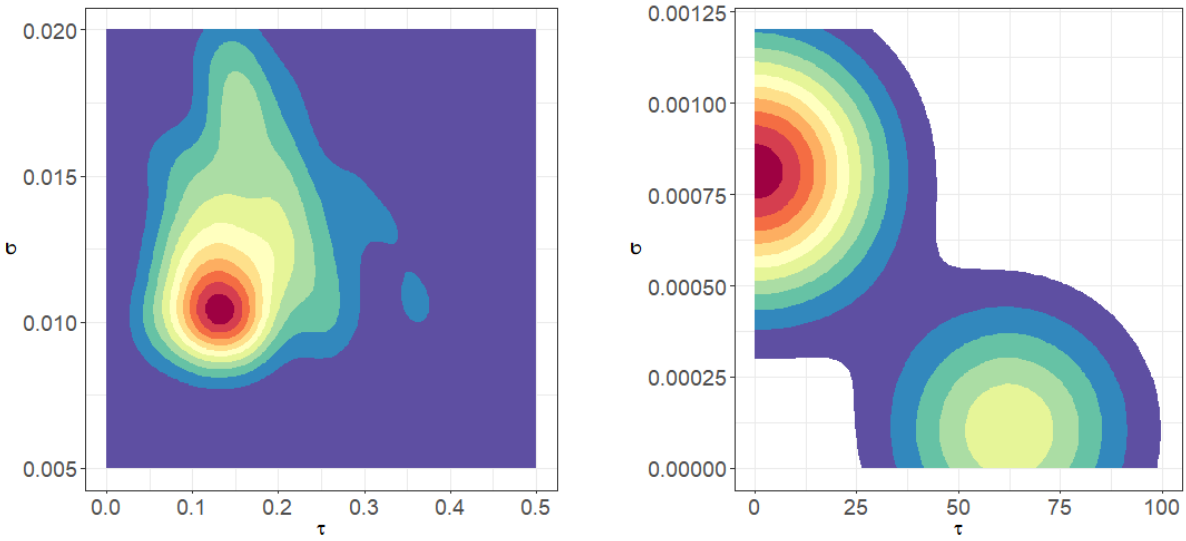
As figuras 35 e 36 apresentam os resultados obtidos no processo de ajuste de hiperparâmetros para cada base de dados considerando o modelo TCSMO-LSSVM. Os resultados do ajuste de hiperparâmetros para o modelo SCG-LSSVM são apresentados nas figuras 37 e 38. Além disso, a Tabela 14 apresenta os valores dos hiperparâmetros ótimos obtidos durante o processo de ajuste.

Tabela 14 – Valores ótimos para cada hiperparâmetro.

Base	**Modelo**	**Hiperparâmetro**	**Valores**
ABA	LSSVM	$[\sigma, \tau]$	$[3.63e^{-2}, 2.82e^{0}]$
ABA	TCSMO-LSSVM	$[\sigma, \tau]$	$[1.00e^{-2}, 1.63e^{-1}]$
ABA	SCG-LSSVM	$[\sigma, \tau]$	$[6.40e^{-3}, 4.45e^{2}]$
MPG	LSSVM	$[\sigma, \tau]$	$[6.60e^{-1}, 1.82e^{0}]$
MPG	TCSMO-LSSVM	$[\sigma, \tau]$	$[3.09e^{0}, 3.82e^{3}]$
MPG	SCG-LSSVM	$[\sigma, \tau]$	$[7.10e^{-2}, 4.88e^{1}]$
CON	LSSVM	$[\sigma, \tau]$	$[1.01e^{-1}, 9.52e^{0}]$
CON	TCSMO-LSSVM	$[\sigma, \tau]$	$[3.80e^{-2}, 1.03e^{2}]$
CON	SCG-LSSVM	$[\sigma, \tau]$	$[4.17e^{-2}, 1.32e^{2}]$
ENE	LSSVM	$[\sigma, \tau]$	$[1.40e^{-2}, 4.27e^{0}]$
ENE	TCSMO-LSSVM	$[\sigma, \tau]$	$[2.05e^{-2}, 1.21e^{3}]$
ENE	SCG-LSSVM	$[\sigma, \tau]$	$[5.92e^{-2}, 3.94e^{2}]$

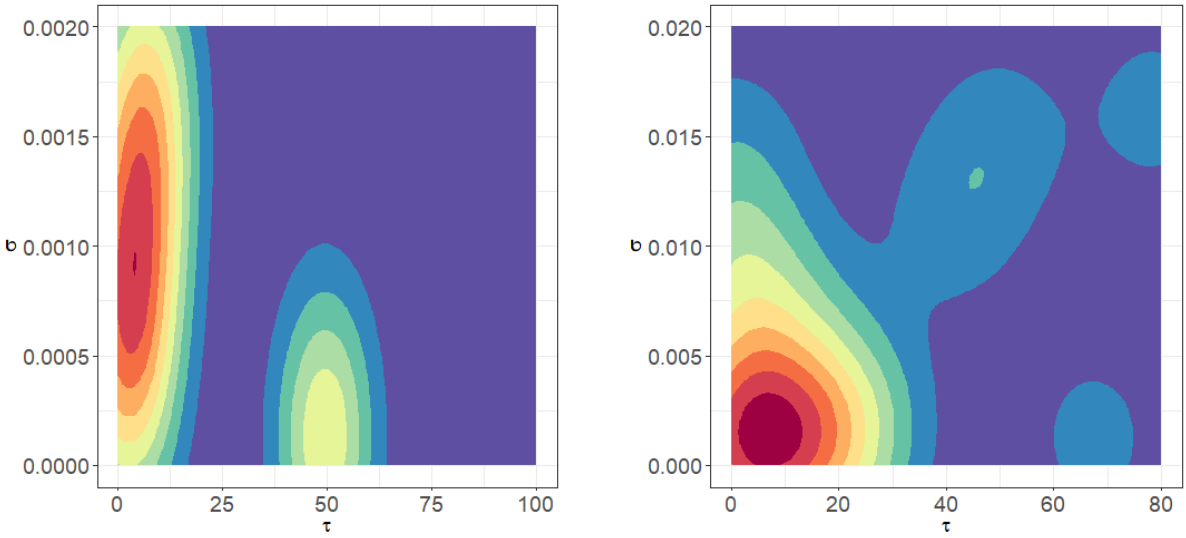
Fonte: Elaborada pelo autor.

Figura 35 – Resultados para o ajuste de hiperparâmetros para as bases ABA (esquerda) e MPG (direita) para o TCSMO-LSSVM.



Fonte: Elaborada pelo autor.

Figura 36 – Resultados para o ajuste de hiperparâmetros para as bases CON (esquerda) e ENE (direita) para o TCSMO-LSSVM.

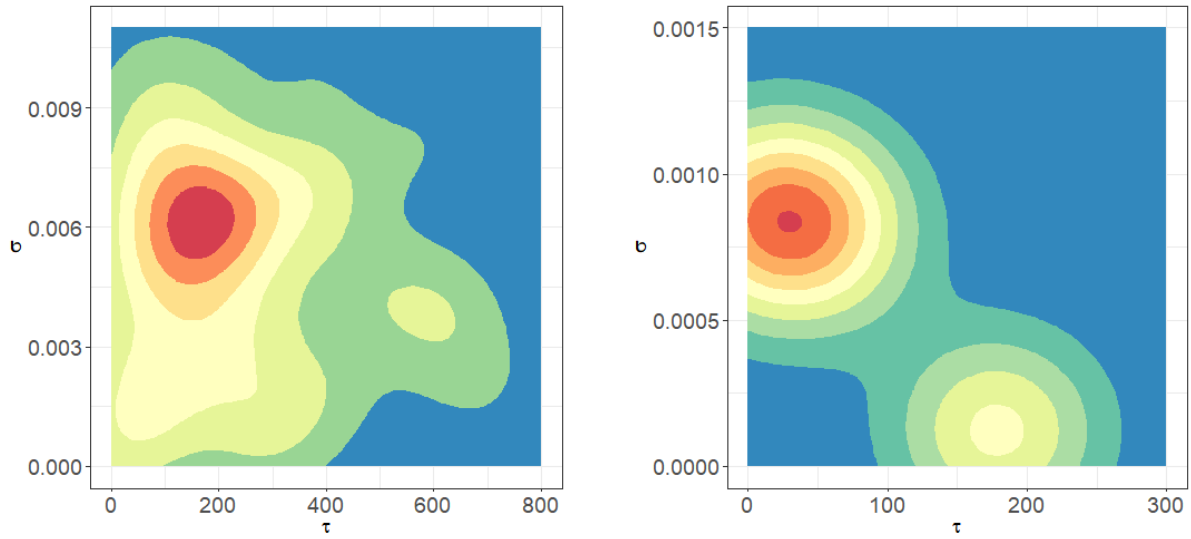


Fonte: Elaborada pelo autor.

6.2.2 Acurácia

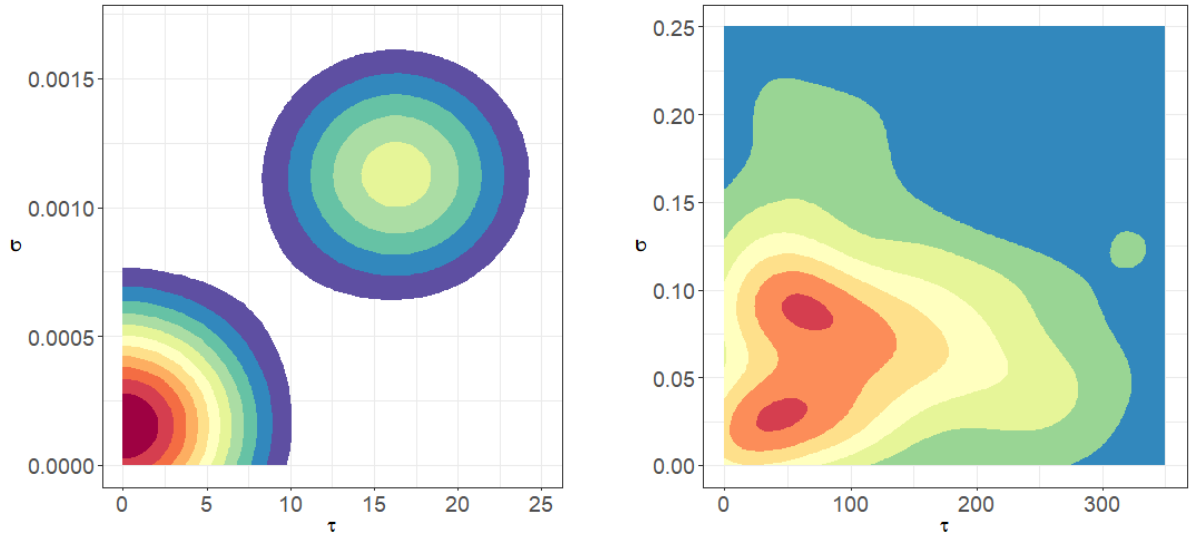
A Tabela 15 apresenta os resultados obtidos pelo LSSVM e para as duas propostas TCSMO-LSSVM e SCG-LSSVM, quando aplicados aos conjuntos de dados reais. Para obter o desempenho dos regressores, foram realizadas 20 rodadas independentes, nas quais calculou-se a média e o desvio padrão para as métricas RMSE e coeficiente de determinação (R^2). Também foram fornecidas as porcentagens de redução no número de vetores-suporte em relação à quantidade do conjunto de vetores de treinamento como dada pela expressão $1 - \frac{\#SV}{\#N}$, com a cardinalidade do conjunto de vetores-suporte dada por $\#SV$ e $\#N$ o número de amostras de

Figura 37 – Resultados para o ajuste de hiperparâmetros para as bases ABA (esquerda) e MPG (direita) para o SCG-LSSVM.



Fonte: Elaborada pelo autor.

Figura 38 – Resultados para o ajuste de hiperparâmetros para as bases CON (esquerda) e ENE (direita) para o SCG-LSSVM.



Fonte: Elaborada pelo autor.

treinamento. Um resumo dos resultados reportados na Tabela 15 é apresentado na Figura 40.

Neste ponto, ressalta-se que foi utilizado o mesmo critério de poda baseado no ganho funcional para esparsificar a solução obtida pelo algoritmo SCG-LSSVM, mostrando que tal procedimento pode ser aplicado em modelos diversos, desde que o treinamento do LSSVM seja realizado no dual. Além disso, pela avaliação da Tabela 15, nota-se o baixo desempenho obtido pelo modelo TCSMO para todas as bases de regressão. De fato, pela observação das Figuras 35 e 36, percebe-se a ocorrência de ótimos locais na etapa de ajuste de hiperparâmetros.

Problemas na convergência se mantiveram quando analisado a variação da métrica

R^2 em termos do percentual de poda variando na faixa de $[0\%, 5\%, 10\%, 15\%, 20\%, 25\%, 30\%]$, o que justifica o uso do percentual de 5% nos resultados apresentados que se seguem, uma vez que, este percentual foi o que forneceu uma melhor capacidade preditiva, embora se tenha baixo nível de esparsidade.

Tabela 15 – Resultados para os regressores treinados de acordo com a metodologia 70/30 usando *kernel* Gaussiano Radial.

Base	**Regressor**	$R^2(\%)$	**RMSE**	**Redução** (%)	**Tempo**(s)
ABA	LSSVM	55.42 ± 2.11	2.16 ± 0.04	0	$3.28e^1 \pm 0.84$
ABA	TCSMO-LSSVM	51.65 ± 3.48	2.40 ± 0.28	5	$2.70e^1 \pm 0.62$
ABA	SCG-LSSVM	90.41 ± 2.0	2.22 ± 0.09	15	$2.90e^1 \pm 0.77$
MPG	LSSVM	85.11 ± 2.01	3.06 ± 0.25	0	$2.28e{-}1 \pm 0.02$
MPG	TCSMO-LSSVM	72.68 ± 3.75	4.56 ± 0.99	5	$2.63e^{-1} \pm 0.02$
MPG	SCG-LSSVM	87.49 ± 1.40	3.05 ± 0.28	15	$2.81e^{-1} \pm 0.28$
CON	LSSVM	84.63 ± 1.35	6.66 ± 0.38	0	$1.85e^0 \pm 0.04$
CON	TCSMO-LSSVM	77.22 ± 8.54	8.11 ± 1.78	5	$2.42^0 \pm 0.03$
CON	SCG-LSSVM	85.63 ± 1.02	6.44 ± 0.31	15	$1.73e^0 \pm 0.04$
ENE	LSSVM	96.10 ± 0.98	1.86 ± 0.19	0	$1.03e^0 \pm 0.03$
ENE	TCSMO-LSSVM	88.41 ± 3.64	3.20 ± 0.63	5	$1.02e^0 \pm 0.01$
ENE	SCG-LSSVM	95.69 ± 1.03	1.95 ± 0.19	15	$9.56e^{-1} \pm 0.03$

Fonte: Elaborada pelo autor.

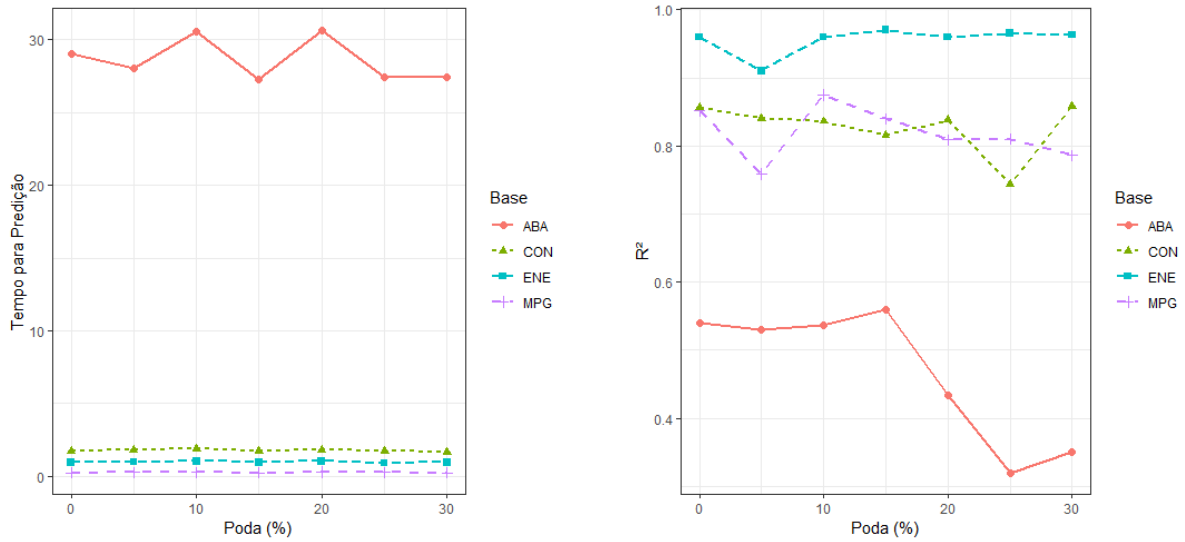
As dificuldades no processo de otimização podem estar relacionadas a etapa de busca do conjunto de trabalho no algoritmo SMO, uma vez que, tal procedimento pode se tornar consideravelmente instável para casos de mal condicionamento da matriz de *kernel*. Testes realizados com o algoritmo CSMO para a tarefa de regressão também ilustram desempenho preditivo ruim o que ratifica a conjectura de que o procedimento de busca esteja dificultando a convergência de tais metodologias. Portanto, futuras investigações e adaptações serão necessárias para garantir uma melhor capacidade preditiva da proposta TCSMO-LSSVM em tarefas de regressão.

Para o algoritmo SCG-LSSVM, os resultados obtidos foram competitivos e em algumas bases de dados superiores ao LSSVM padrão, o que ratifica a fundamentação teórica envolvida no seu desenvolvimento, além de, demonstrar flexibilidade em tratar tanto problemas de classificação como regressão. Destaca-se também a rápida convergência, tornando o treinamento mais rápido que o LSSVM para a maioria das bases analisadas.

Ressalta-se que a esparsificação das soluções obtidas pelo SCG-LSSVM foi realizada pelo esquema de poda baseado no ganho funcional e, portanto, é necessário definir um percentual de poda ótima que concilia o *dilema* entre ter um maior nível de esparsidade sem grande perdas de desempenho preditivo por parte do modelo. Assim como na análise

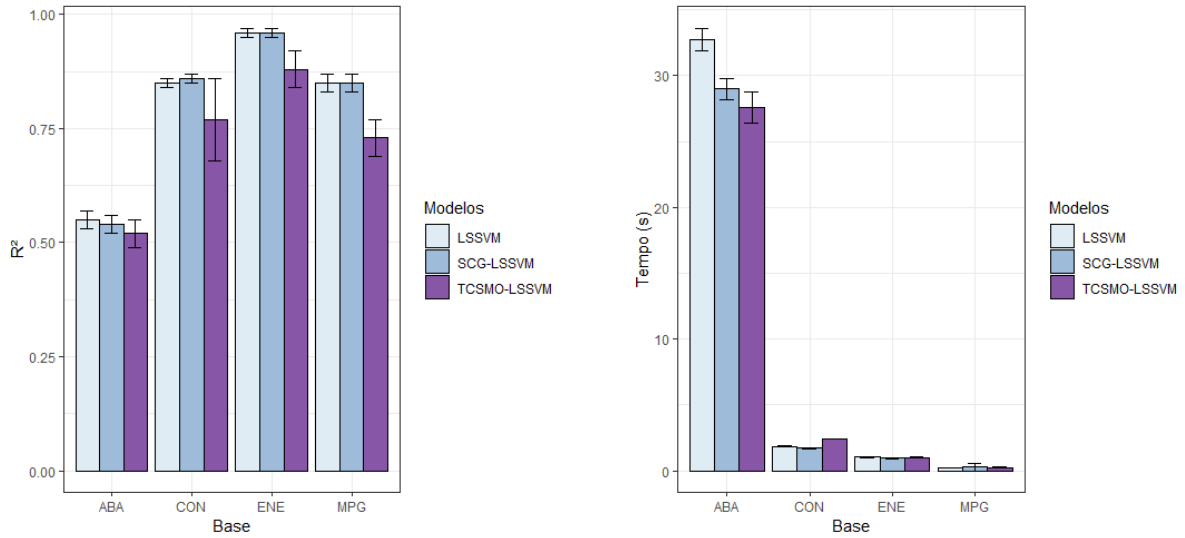
de bases de classificação, definiu-se um *grid* de valores para o porcentagem de poda indo de 0% até 30% em passos de 5%, ou seja, foram considerados os valores de percentual de poda de $[0\%, 5\%, 10\%, 15\%, 20\%, 25\%, 30\%]$ para o SCG-LSSVM, avaliando-se o R^2, bem como, o tempo de processamento do estágio de predição. Os resultados são reportados na Figura 39. Nota-se que o percentual de 15% permite um nível de esparsidade sem uma brusca queda na performance preditiva do modelo.

Figura 39 – R^2 e tempo de predição em termos de porcentagem de poda no SCG-LSSVM para os conjuntos de dados ABA, MPG, CON e ENE.



Fonte: Elaborada pelo autor.

Figura 40 – R^2 e tempo de treinamento para cada modelo nos conjuntos de dados ABA, MPG, CON e ENE.

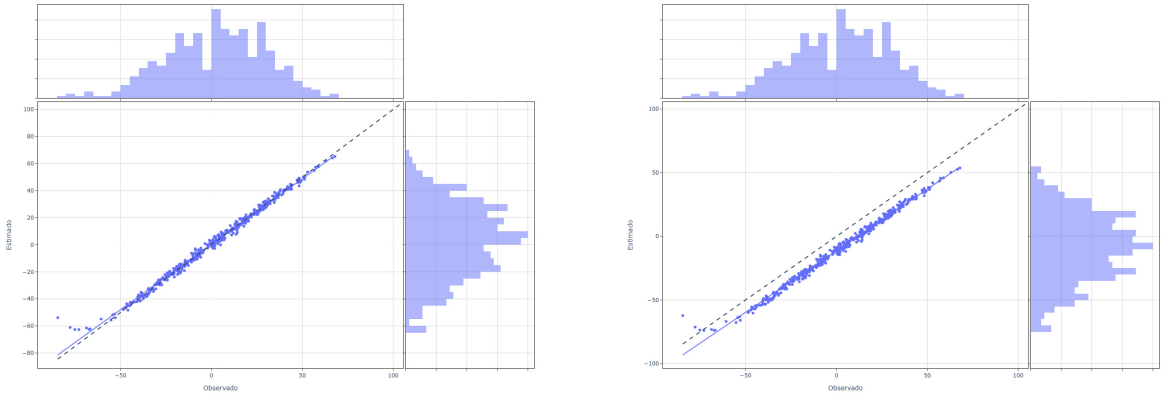


Fonte: Elaborada pelo autor.

6.2.3 Análise Qualitativa Visual

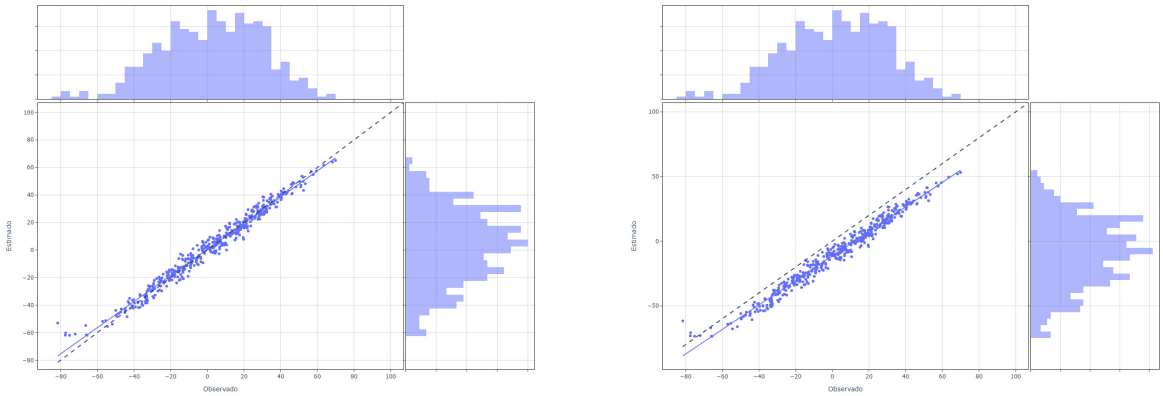
Com o intuito de desenvolver uma análise qualitativa sobre a capacidade preditiva das propostas, foram desenvolvidos duas bases de dados artificiais geradas de funções conhecidas. No primeiro caso, foi considerado uma distribuição de ruído uniforme sobre uma curva gerada aleatoriamente utilizando a biblioteca *sklearn*, já no segundo considerou-se o mesmo procedimento com um nível de ruído ampliado ao dobro do primeiro cenário, em ambos os casos as amostras apresentam volumetria de 1500 observações. Os resultados apresentados nas Figuras 41 e 42 confirmam que o bom desempenho preditivo do modelo SCG-LSSVM e a dificuldade de convergência por parte do TCSMO-LSSVM.

Figura 41 – Comparativo ente os valores observado e estimados para a primeira base sintética considerando o SCG-LSSVM (esquerda) e TCSMO-LSSVM (direita).



Fonte: Elaborada pelo autor.

Figura 42 – Comparativo ente os valores observado e estimados para a segunda base sintética considerando o SCG-LSSVM (esquerda) e TCSMO-LSSVM (direita).



Fonte: Elaborada pelo autor.

Assim a proposta SCG-LSSVM, apresenta-se como uma nova proposta com potencial

positivo para uso em tarefas de regressão, principalmente em cenários com bases de dados grandes, devido a sua boa capacidade discriminativa, rápido treinamento e esparsidade permitida pelo procedimento de poda. Relativo a proposta TCSMO-LSSVM melhorias e adaptações no FGWSS devem ser realizadas com o intuito de melhorar a convergência do algoritmo dentro do processo de treinamento. Possíveis medidas a serem investigadas, seriam a adição de termos de momento na atualização da variável com o intuito de forçar que soluções não ocorram em ótimos locais, bem como, alterações na formulação da solução do problema dual, como a adição de derivadas de ordem fracionárias, visando uma suavização do processo de otimização e evitando a ocorrência de pontos estacionários.

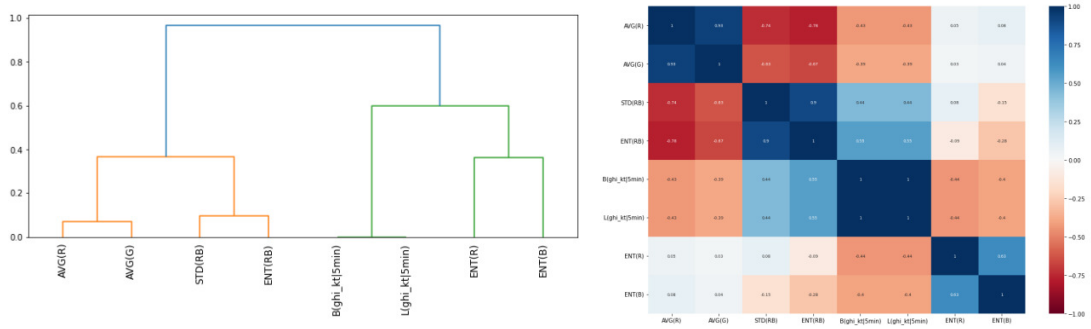
6.3 Resultados para a Base de Folsom, CA

Nesta seção são apresentados os resultados para as previsões GHI e DNI considerando as bases resultantes do processo de seleção de atributos com apenas 5 preditores para ambos os casos, GHI e DNI. Para o treinamento dos modelos XGBoost e LightGBM foi empregado otimização Bayesiana de uma forma similar aos algoritmos propostos. O modelo GMDH com abordagem combinatória, utiliza uma estratégia auto-organizável e consegue se ajustar aos dados sem necessitar de ajuste de hiperparâmetros.

Para a etapa de seleção de atributos baseada na análise bivariada, utilizando a correlação de *Spearman* em conjunto com um *clustering* hierárquico com o intuito de remover variáveis correlacionadas, sem que ocorra a retirada de preditores com alto poder discriminativo para as previsões GHI e DNI. Primeiramente, realiza-se o *clustering* hierárquico utilizando como medida de dissimilaridade o valor $1 - corr(x_i, x_j)$, desta forma elementos de um mesmo *cluster* representam variáveis correlacionadas. A Figura 43 apresenta os *clusters* resultantes deste procedimento, em conjunto com a matriz de correlação realçando os agrupamentos formados considerando todas as variáveis de entrada presentes na base.

Uma vez detectado cada agrupamento com as variáveis correlacionadas, foram removidas para cada um, aquelas variáveis que apresentavam uma correlação de *Spearman* inferior a 0.4 em valor absoluto com a variável de saída, ou seja, que satisfazem a condição $|corr(x_i, y)| < 0.4$. Este procedimento garante que para cada *cluster*, somente aquelas variáveis com menor impacto preditivo para variável de saída serão removidas. Para previsões GHI, as variáveis restantes após este filtro, foram $B(ghi_kt|5min), L(ghi_kt|5min)$, AVG(R), ENT(R), AVG(G), ENT(B), STD(RB), ENT(RB).

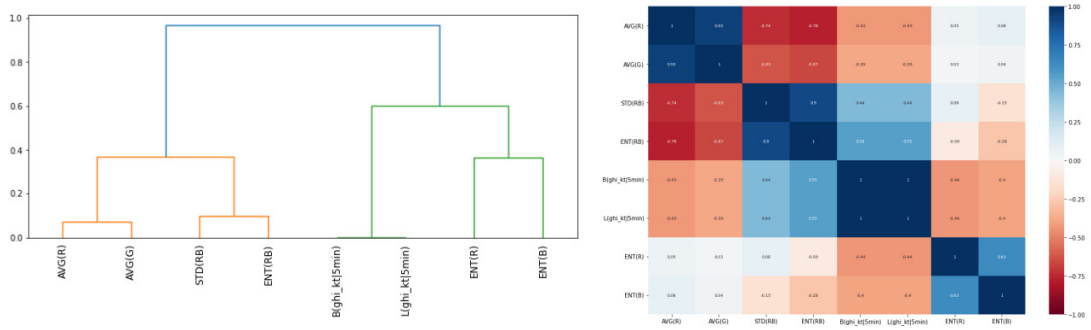
Figura 43 – *Clusters* resultantes e correspondente matriz de correlação.



Fonte: Elaborada pelo autor.

A Figura 44 apresenta os resultados de *clustering* hierárquico em conjunto com sua matriz de correlação onde fica evidenciado, que apesar de ainda existir alguns preditores correlacionados, os mesmos não foram removidos devido ao seu alto poder discriminativo para a variável de saída, no caso kt_{GHI}

Figura 44 – *Clusters* resultantes e correspondente matriz de correlação para as variáveis resultantes na previsão GHI.

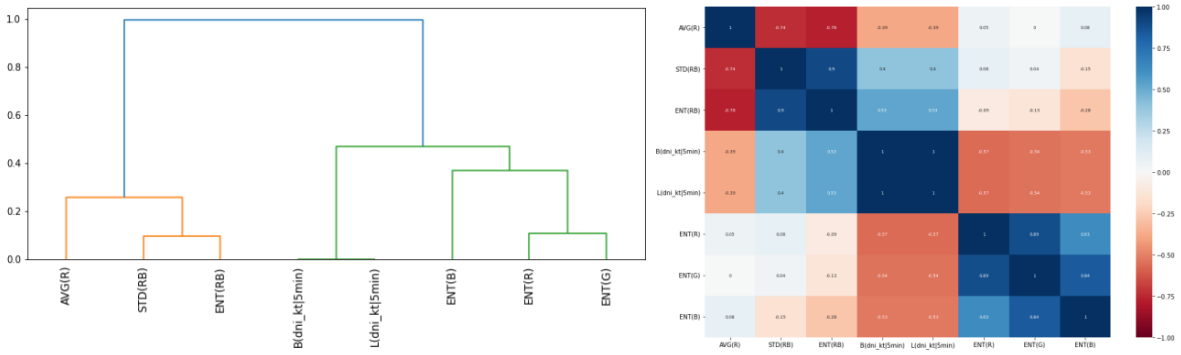


Fonte: Elaborada pelo autor.

A Figura 45 apresenta os mesmos resultados para o caso de previsões DNI. Neste caso, as variáveis restantes foram $B(dni_{kt|5min}), L(dni_{kt|5min})$, AVG(R), ENT(R), ENT(G), ENT(B), STD(RB), ENT(RB).

Por fim, um procedimento de RFE foi realizado com o intuito de reduzir ainda mais a dimensionalidade do problema. Destaca-se aqui, que foi utilizado o modelo XGBoost para a realização do RFE em conjunto com uma validação cruzada 5 *folds* para avaliação do desempenho de cada modelo a medida em que as variáveis iam sendo removidas. Por fim,

Figura 45 – *Clusters* resultantes e correspondente matriz de correlação para as variáveis resultantes na previsão DNI.



Fonte: Elaborada pelo autor.

configurou-se o algoritmo para que o número de variáveis resultante fosse de 5.

Ao executar o *pipeline* descrito na Figura 15, o conjunto de dados resultante teve 5 preditores: **B**, **V**, μ_R, H_R e H_{RB}, no contexto de previsão de GHI, e as variáveis **B**, μ, H_R, H_{RB} e H_G para previsão de DNI. Os resultados obtidos considerando apenas estas variáveis são apresentados nas Tabelas 16 e 17 para previsões GHI e DNI e resumidos de uma forma esquemática pela Figura 46. Destaca-se que devido aos problemas de convergência notados no TCSMO-LSSVM em problemas de regressão, a análise para esta base de dados foi realizada considerando apenas a proposta SCG-LSSVM.

Tabela 16 – Resultados para previsões GHI.

Modelos	**Métricas**	**5 min**	**10 min**	**15 min**	**20 min**	**25 min**	**30 min**
GMDH	RMSE	50.44	62.11	67.89	71.39	73.92	76.44
GMDH	R^2	0.97	0.95	0.94	0.94	0.93	0.93
LightGBM	RMSE	47.49	57.98	62.97	66.40	68.96	71.51
LightGBM	R^2	0.97	0.96	0.95	0.95	0.94	0.94
XGBoost	RMSE	47.25	57.15	62.28	65.71	68.54	71.04
XGBoost	R^2	0.97	0.96	0.95	0.95	0.94	0.94
SCG-LSSVM	RMSE	94.01	98.28	124.88	128.54	129.67	132.06
SCG-LSSVM	R^2	0.88	0.86	0.70	0.69	0.68	0.67

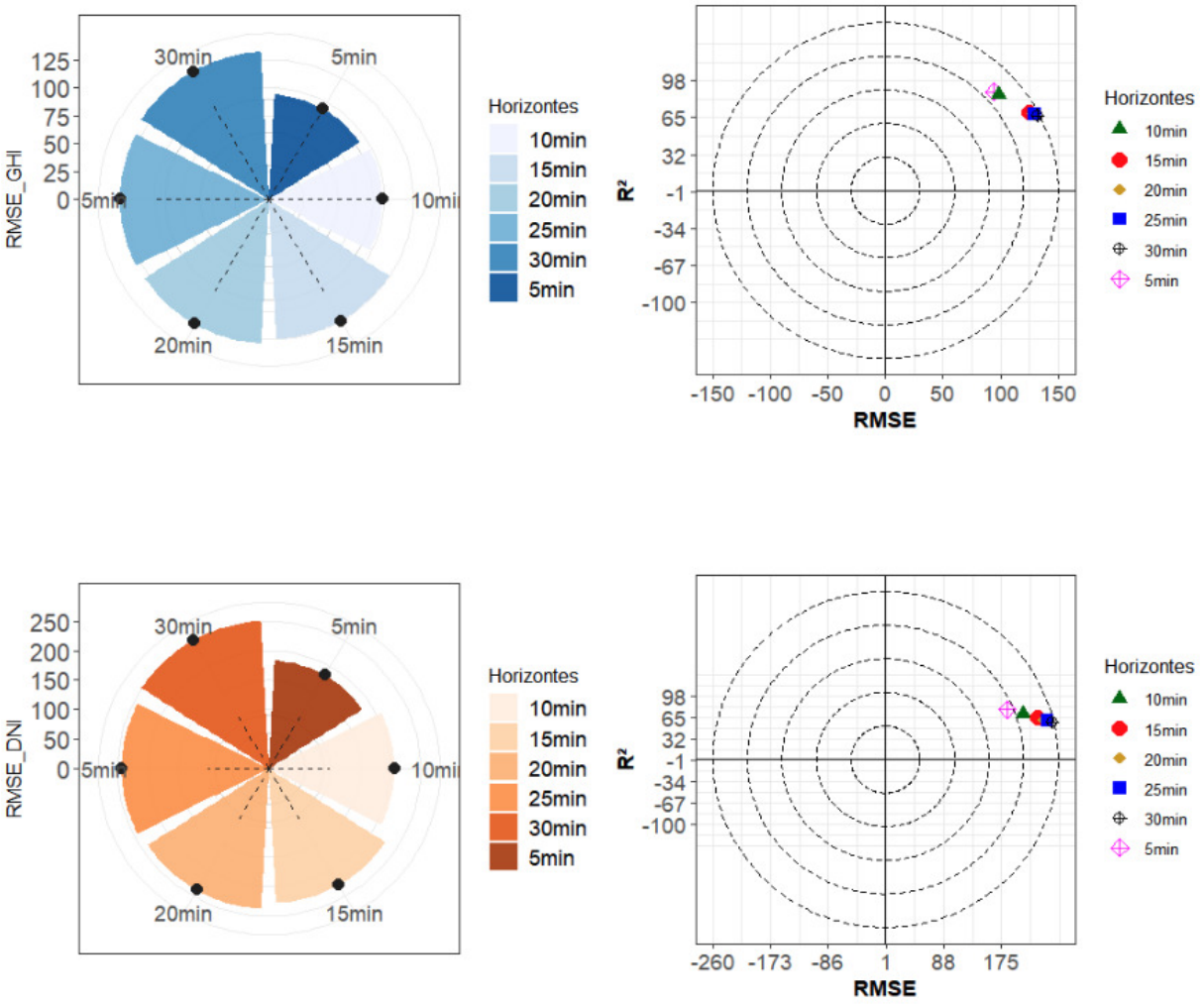
Fonte: Elaborada pelo autor.

Tabela 17 – Resultados para previsões DNI.

Modelos	Métricas	5 min	10 min	15 min	20 min	25 min	30 min
GMDH	RMSE	86.16	109.10	123.44	133.60	141.18	148.11
GMDH	R^2	0.94	0.90	0.87	0.85	0.84	0.83
LightGBM	RMSE	82.39	105.06	117.51	126.44	133.45	139.83
LightGBM	R^2	0.94	0.91	0.89	0.87	0.86	0.84
XGBoost	RMSE	81.72	103.14	116.60	126.00	133.52	139.49
XGBoost	R^2	0.94	0.91	0.89	0.87	0.86	0.84
SCG-LSSVM	RMSE	183.53	207.53	228.85	238.27	243.51	250.93
SCG-LSSVM	R^2	0.77	0.71	0.65	0.62	0.60	0.58

Fonte: Elaborada pelo autor.

Figura 46 – Aumento nos valores de RMSE ao longo dos horizontes de previsão.



Fonte: Elaborada pelo autor.

Pela observação dos resultados apresentados nas Tabelas 16 e 17, nota-se que a performance preditiva do SCG-LSSVM é inferior aos demais modelos para a base considerada, o que era esperado dada a elevada capacidade generalização em dados estruturados de modelos

de árvores de decisão e de rede neurais artificiais. Outro ponto a ser considerado está na piora nos valores das métricas de desempenho com o aumento do horizonte de previsão, já que com isto o nível de incerteza na previsão da saída é ampliado o que é convergente com os resultados obtidos em Pedro e Coimbra (2015), Pedro *et al.* (2019).

Além disso, outro comportamento esperado é um pior desempenho preditivo dos modelos na tarefa de previsão DNI relativo a GHI. Esse fato é comum e esperado, uma vez que o GHI contabiliza toda a radiação que atinge o solo, vinda de todas as direções, enquanto, por outro lado, o DNI é uma medição fortemente direcional, e até mesmo pequenas nuvens podem bloquear o Sol, levando a erros nas estimativas.

De uma forma geral, nota-se que o modelo SCG-LSSVM apresentou resultados consistentes e com comportamento convergente com o que é reportado na literatura sobre previsão de irradiância de curto prazo. Além disso, dado seu rápido treinamento com esparsidade promovida pela poda, esta metodologia pode ser extremamente adequada para o desenvolvimento de dispositivos de previsão embarcados com limitações de *hardware* como plataformas *Arduino* e *Raspberry Pi*.

Neste capítulo, extensivas simulações numéricas sobre bases de dado reais e artificiais, considerando tanto problemas de classificação binária como de estimação de funções foram realizadas com o intuito de validar as metodologias propostas. Cenários de baixa e de alta volumetria foram considerados, análise visual da fronteira de decisão em classificação binária e verificação do desempenho preditivo do SCG-LSSVM sobre uma base real de previsão de irradiância solar de curto prazo. Notou-se que as duas propostas apresentaram desempenho preditivo competitivo quando comparado a outros métodos mais bem consolidados, com a vantagem de rápido treinamento e esparsidade promovida sobre vetor ótimo de multiplicadores de Lagrange.

Para problemas de regressão, a proposta SCG-LSSVM se mostrou mais estável que o TCSMO-LSSVM em todas as bases consideradas, resultando em uma metodologia alternativa para a obtenção de LSSVRs esparsos e de rápido treinamento. Um maior detalhamento das conclusões obtidas será apresentada no próximo capítulo, no qual ainda tem-se a apresentação de dificuldades inerentes de cada proposta, bem como, o delineamento de sugestões de melhorias para pesquisas futuras.

7 CONCLUSÕES E TRABALHOS FUTUROS

Nesta tese, foi abordado o problema da carência de eparsidade nas soluções promovidas por modelos LSSVMs, o que caracteriza-se por ser sua principal deficiência, uma vez que, a consideração de todos os padrões de treinamento como vetores-suporte impacta diretamente na fase predição do modelo tornando-o ineficiente em cenários de processamento de grandes bases de dados. Com isso, foram propostos dois novos algoritmos para o treinamento rápido e esparso de modelos LSSVMs, em que, ambos solucionam o problema dual diretamente de tais modelos.

7.1 Conclusões

As duas novas propostas tratam do problema dual resultante da formulação do treinamento de LSSVMs. No TCSMO-LSSVM, emprega-se o algoritmo SMO com uma nova direção de descida de três termos, que permite um maior ganho funcional por iteração, acelerando a etapa de treinamento, mas com uma solução ótima densa. Para conseguir esparsidade uma metodologia de poda com tamanho fixado foi adotada, em que padrões com menores contribuições para o ganho funcional do problema de otimização foram removidos.

Para o SCG-LSSVM, emprega-se um novo método do gradiente conjugado espectral com um novo esquema de escolha dos parâmetros espectral e conjugado que favorece o desenvolvimento de uma nova direção de busca que satisfaz a propriedade espectral e a condição de descida simultaneamente, favorecendo alto ganho funcional a cada iteração. Além disso, neste novo algoritmo, emprega-se o uso de informação de segunda ordem por meio da aproximação BFGS para a Hessiana da função objetivo dual o que contribui para uma rápida convergência sem grande custos de processamento por iteração.

Os dois novos métodos propostos foram validados por meio de diversas simulações computacionais tanto em bases de classificação como em bases de regressão. Em ambos os casos, foram consideradas as análises do desempenho preditivo e tempo de processamento no treino para bases pequenas, análise da etapa de *tuning* de hiperparâmetros, análise qualitativa do desempenho por meio da fronteira de decisão e também por meio de gráficos de dispersão entre valores medidos e estimados para o caso de regressão, bem como, avaliação do desempenho em bases de dados grandes.

Primeiramente, ambas as propostas apresentaram desempenho preditivo competitivo quando comparado ao LSSVM padrão e a outras variantes esparsas já consolidadas com a

vantagem de fornecer rápido treinamento e esparsidade da solução quando aplicados sobre as bases que abordam a tarefa de classificação binária. Neste cenário, o TCSMO-LSSVM conseguiu um maior destaque do que SCG-LSSVM, com uma melhor acurácia e tempos de treinamento um pouco menores, embora tenha apresentado para algumas bases, problemas relacionados à convergência na etapa de *tuning* de hiperparâmetros.

O SCG-LSSVM apresentou maior robustez na etapa de *tuning*, além disso, esta proposta apresenta um procedimento de poda baseado na distância de um determinado padrão ao hiperplano de decisão, de tal forma, que não é necessário por parte do usuário definir um percentual de poda, ou seja, esta metodologia ajusta o nível de poda de uma forma automática sendo, portanto, uma metodologia de tamanho variável.

Em ambos os casos, as fronteiras de decisão geradas se mostraram condizentes com aquela desenvolvida pelo modelo de *benchmarking* IP-LSSVM em todas as bases sintéticas utilizadas para essa análise. Também vale destacar a boa capacidade preditiva e o treinamento em tempo hábil quando as propostas foram aplicadas sobre grandes bases de dados em problemas de classificação binária. Também é interessante notar que apenas as metodologias que empregam a abordagem dual foram capazes de tratar com tais bases, nos demais métodos, problemas relacionados a alocação de memória e tempo para o processamento foram observados.

Para as bases de regressão, o TCSMO-LSSVM apresentou desempenho inferior aos demais modelos de comparação. Este comportamento se repetiu para o método CSMO-LSSVM, enquanto para o SCG-LSSVM o desempenho se manteve robusto e competitivo com rápido treinamento e esparsidade garantida por poda de tamanho fixado. Com isso, conjectura-se que o problema esteja na etapa de seleção do conjunto de trabalho, FGWSS, para o algoritmo SMO, que necessita de novas adaptações para o cenário de regressão.

O SCG-LSSVM foi avaliado frente a modelos de estado da arte em tarefas de aprendizado em bases de dados estruturadas. Esta análise comparativa foi realizada sobre uma grande base de dados reais obtida em um problema de previsão de recurso solar de curto prazo. Foram realizadas previsões em seis horizontes de 5 minutos à 30 minutos em passos de 5, tanto para GHI como para DNI, em que os resultados indicaram uma capacidade preditiva inferior da proposta quando comparado aos demais modelos utilizados na comparação, embora os resultados obtidos tenham sido consistentes com o que é reportado na literatura sobre previsão de irradiância solar de curto prazo.

Por fim, realizando diversas simulações computacionais sobre bases de classificação

e regressão, percebe-se que as duas propostas apresentadas possuem desempenho preditivo competitivo em tarefas de classificação binária com rápido treinamento e esparsidade na solução ótima dos multiplicadores de Lagrange, já para as bases de regressão a proposta TCSMO-LSSVM apresentou dificuldades relacionadas a convergência na etapa do FGWSS, já o SCG-LSSVM forneceu capacidade preditiva competitiva em ambas as tarefas de classificação binária e regressão, sendo portanto, uma interessante alternativa para métodos de *kernel* com rápido treinamento e garantias de esparsidade.

7.2 Direções Futuras

Dentre os possíveis direcionamentos que podem ser adotados para a melhoria desta pesquisa, destacam-se os seguintes pontos:

- Melhorias sobre a metodologia FGWSS para a seleção do conjunto de trabalho no algoritmo SMO devem ser realizadas com o intuito de tornar a primeira proposta mais robusta a diferentes bases tanto considerando tarefas de classificação como regressão;
- Encapsulamento de todos os *scripts* desenvolvidos na forma de uma *lib*, considerando todos os padrões e boas práticas de desenvolvimento de *software* para que futuros usuários possam utilizar e melhorar as funcionalidades já desenvolvidas;
- Adaptar e melhorar o procedimento de poda para o TCSMO-LSSVM, fazendo com que a mesma seja de tamanho variável, contribuindo para a redução no número de hiperparâmetros do modelo;
- Complementar a formulação dos algoritmos para que os mesmos sejam robustos à ocorrência de *outliers* nos dados de entrada para o modelo LSSVM utilizando estimação M ou mínimos quadrados ponderados, por exemplo;
- Formular versões das duas propostas considerando o processamento de dados em *stream* para cenários de aprendizado *online* com limitações de *hardware*;
- Utilizando todos esses avanços, embarcar todas as propostas em um dispositivo de previsão de irradiância solar de baixo custo e avaliar sua viabilidade em um cenário real de previsão de recurso solar.

REFERÊNCIAS

ANASTASAKIS, L.; MORT, N. **The development of self-organization techniques in modelling: a review of the group method of data handling (GMDH)**. Sheffield: University of Sheffield, Department of Automatic Control and Systems Engineering, 2001. Research report.

BABAIE-KAFAKI, S. On optimality of the parameters of self-scaling memoryless quasi-newton updating formulae. **Journal of Optimization Theory and Applications**, Springer, v. 167, p. 91–101, 2015.

BARZILAI, J.; BORWEIN, J. M. Two-point step size gradient methods. **IMA journal of numerical analysis**, Oxford University Press, v. 8, n. 1, p. 141–148, 1988.

BAY, S. D.; KIBLER, D.; PAZZANI, M. J.; SMYTH, P. The uci kdd archive of large data sets for data mining research and experimentation. **ACM SIGKDD explorations newsletter**, ACM New York, NY, USA, v. 2, n. 2, p. 81–85, 2000.

BEALE, E. A derivation of conjugate-gradients. **Numerical methods for non-linear optimization**, Academic Press, 1972.

BEN-ISRAEL, A.; GREVILLE, T. N. **Generalized inverses:** theory and applications. [*S. l.*]: Springer Science & Business Media, 2006.

BIRGIN, E. G.; MARTÍNEZ, J. M. A spectral conjugate gradient method for unconstrained optimization. **Applied Mathematics & Optimization**, Springer, v. 43, n. 2, p. 117–128, 2001.

BREIMAN, L.; FRIEDMAN, J.; OLSHEN, R. A.; STONE, C. J. **Classification and regression trees**. [*S. l.*]: Chapman and Hall/CRC, 2017.

BROYDEN, C. Quasi-newton, or modification methods. *In:* Numerical solution of systems of nonlinear algebraic equations. [*S. l.*]: Elsevier, 1973. p. 241–280.

BROYDEN, C. G.; JR, J. E. D.; MORÉ, J. J. On the local and superlinear convergence of quasi-newton methods. **IMA Journal of Applied Mathematics**, Oxford University Press, v. 12, n. 3, p. 223–245, 1973.

BRUNTON, S. L.; ZOLMAN, N.; KUTZ, J. N.; FASEL, U. Machine learning for sparse nonlinear modeling and control. **Annual Review of Control, Robotics, and Autonomous Systems**, Annual Reviews, v. 8, 2025.

CARVALHO, B. P. R.; BRAGA, A. P. Ip-lssvm: a two-step sparse classifier. **Pattern Recognition Letters**, Elsevier, v. 30, n. 16, p. 1507–1515, 2009.

CHEN, F.; LI, S.; HAN, J.; REN, F.; YANG, Z. Review of lightweight deep convolutional neural networks. **Archives of Computational Methods in Engineering**, Springer, v. 31, n. 4, p. 1915–1937, 2024.

CHEN, H.; LV, C. Online learning-informed feedforward-feedback controller synthesis for path tracking of autonomous vehicles. **IEEE Transactions on Intelligent Vehicles**, IEEE, v. 8, n. 4, p. 2759–2769, 2022.

CHUA, K. S. Efficient computations for large least square support vector machine classifiers. **Pattern Recognition Letters**, Elsevier, v. 24, n. 1-3, p. 75–80, 2003.

CORTES, C.; VAPNIK, V. Support vector machine. **Machine learning**, v. 20, n. 3, p. 273–297, 1995.

CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support vector machines and other kernel-based learning methods**. [*S. l.*]: Cambridge university press, 2000.

DAI, Y.-H.; YUAN, Y. A nonlinear conjugate gradient method with a strong global convergence property. **SIAM Journal on optimization**, SIAM, v. 10, n. 1, p. 177–182, 1999.

DA'U, A.; SALIM, N. Recommendation system based on deep learning methods: a systematic review and new directions. **Artificial Intelligence Review**, Springer, v. 53, n. 4, p. 2709–2748, 2020.

DERVIŞ, H. Bibliometric analysis using bibliometrix an r package. **Journal of scientometric research**, v. 8, n. 3, p. 156–160, 2019.

DIAS, M. L. D.; MAIA, Á. N.; NETO, A. R. d. R.; GOMES, J. P. P. Parsimonious minimal learning machine via multiresponse sparse regression. **International journal of neural systems**, World Scientific, v. 30, n. 05, p. 2050023, 2020.

DIAS, M. L. D.; NETO, A. R. R. Training soft margin support vector machines by simulated annealing: a dual approach. **Expert Systems with Applications**, Elsevier, v. 87, p. 157–169, 2017.

DIAS, M. L. D.; SOUSA, L. S. d.; NETO, A. R. d. R.; FREIRE, A. L. Fixed-size extreme learning machines through simulated annealing. **Neural Processing Letters**, Springer, v. 48, n. 1, p. 135–151, 2018.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of Machine Learning Research**, v. 12, n. 7, 2011.

EL-AMARTY, N.; FADILI, H. E.; BENNANI, S. D. Accurate short-term solar irradiance forecasting with tinyml on edge device. *In:* **INTERNATIONAL CONFERENCE ON CIRCUIT, SYSTEMS AND COMMUNICATION (ICCSC)**. Fez, Marrocos: IEEE, 2024. p. 1–6. Trabalho apresentado no INTERNATIONAL CONFERENCE ON CIRCUIT, SYSTEMS AND COMMUNICATION (ICCSC), 2024, [Fez, Marrocos].

ELKINS, J. G. **Online learning for adaptive control**: Stable learning and control for aerospace and robotics. [*S. l.*]: The University of Alabama in Huntsville, 2024.

FAGHIH, A.; RINELLI, M.; BAREL, M. V.; VANDEBRIL, R.; VERMEIREN, R. Krylov and core transformation algorithms for an inverse eigenvalue problem to compute recurrences of multiple orthogonal polynomials. **arXiv preprint arXiv:2506.19796**, 2025. Disponível em: https://arxiv.org/abs/2506.19796. Acesso em: 28/10/2025.

FAN, R.-E.; CHEN, P.-H.; LIN, C.-J.; JOACHIMS, T. Working set selection using second order information for training support vector machines. **Journal of Machine Learning Research**, v. 6, n. 12, 2005.

FLORÊNCIO, J. A. V.; OLIVEIRA, S. A. F.; GOMES, J. P. P.; NETO, A. R. d. R. A new perspective for minimal learning machines: a lightweight approach. **Neurocomputing**, Elsevier, v. 401, p. 308–319, 2020.

FRAZIER, P. I. Bayesian optimization. *In:* Recent advances in optimization and modeling of contemporary problems. [*S. l.*]: Informs, 2018. p. 255–278.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. [*S. l.*]: "O'Reilly Media, Inc.", 2022.

GOLUB, G. H.; HANSEN, P. C.; O'LEARY, D. P. Tikhonov regularization and total least squares. **SIAM journal on matrix analysis and applications**, SIAM, v. 21, n. 1, p. 185–194, 1999.

GOLUB, G. H.; LOAN, C. F. V. **Matrix computations**. [*S. l.*]: JHU press, 2013.

GRININ, L.; GRININ, A. Macroevolution of technology. *In:* GRININ, L. E.; KOROTAYEV, A. V. (Ed.). **Evolution: development within Big History, evolutionary and world-system paradigms**. Volgograd: Uchitel Publishing House, 2013. p. 143–178.

HALLER, G.; KASZÁS, B. Data-driven linearization of dynamical systems. **Nonlinear Dynamics**, Springer, v. 112, n. 21, p. 18639–18663, 2024.

HILBERT, M. Digital technology and social change: the digital transformation of society from a historical perspective. **Dialogues in clinical neuroscience**, Taylor & Francis, v. 22, n. 2, p. 189–194, 2020.

HMEDE, R.; CHAPELLE, F.; LAPUSTA, Y. Review of neural network modeling of shape memory alloys. **Sensors**, MDPI, v. 22, n. 15, p. 5610, 2022.

HMEDE, R.; CHAPELLE, F.; LAPUSTA, Y. Review of neural network modeling of shape memory alloys. **Sensors**, MDPI, v. 22, n. 15, p. 5610, 2022.

HO, J.; SALIMANS, T.; GRITSENKO, A.; CHAN, W.; NOROUZI, M.; FLEET, D. J. Video diffusion models. **Advances in Neural Information Processing Systems**, v. 35, p. 8633–8646, 2022.

HOVELL, K. C. **Deep reinforcement learning as guidance for aerospace robotics**. Tese (Doutorado em Engenharia Aeroespacial) – Carleton University, Ottawa, 2022.

INEICHEN, P.; PEREZ, R. A new airmass independent formulation for the linke turbidity coefficient. **Solar Energy**, Elsevier, v. 73, n. 3, p. 151–157, 2002.

IZMAILOV, A.; SOLODOV, M. **Otimização, volume 2:** métodos computacionais. [*S. l.*]: IMPA, 2007.

JIANG, M. A review of the impacts of industrial revolutions in world history. **Communications in Humanities Research**, v. 39, n. 1, p. 234–239, 2024.

JIAO, L.; BO, L.; WANG, L. Fast sparse approximation for least squares support vector machine. **IEEE Transactions on Neural Networks**, IEEE, v. 18, n. 3, p. 685–697, 2007.

JIAO, L.; BO, L.; WANG, L. Fast sparse approximation for least squares support vector machine. **IEEE Transactions on Neural Networks**, IEEE, v. 18, n. 3, p. 685–697, 2007.

KOSTOPOULOS, A. E.; GRAPSA, T. Self-scaled conjugate gradient training algorithms. **Neurocomputing**, Elsevier, v. 72, n. 13-15, p. 3000–3019, 2009.

KURBIEL, T.; KHALEGHIAN, S. Training of deep neural networks based on distance measures using rmsprop. **arXiv preprint arXiv:1708.01911**, 2017. Disponível em: https://arxiv.org/abs/1708.01911. Acesso em: 27/04/2025.

LEAO, D. A.; NETO, A. R. R. Um novo método de poda iterativo de máquinas de vetores-suporte de mínimos quadrados. *In:* **CONGRESSO BRASILEIRO DE AUTOMÁTICA (CBA)**. Fortaleza, Brasil: SBA (Sociedade Brasileira de Automática), 2022. v. 3, n. 1. Trabalho apresentado no CONGRESSO BRASILEIRO DE AUTOMÁTICA (CBA), 2022, [Fortaleza, Brasil]. ID do Artigo: 3666. Disponível em: https://doi.org/10.20906/CBA2022/3666.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. **Quarterly of applied mathematics**, v. 2, n. 2, p. 164–168, 1944.

LI, B.; SONG, S.; LI, K. A fast iterative single data approach to training unconstrained least squares support vector machines. **Neurocomputing**, Elsevier, v. 115, p. 31–38, 2013.

LI, F.; YANG, Y. Analysis of recursive feature elimination methods. *In:* **PROCEEDINGS OF THE 28TH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL**. Salvador, Brasil: ACM (Association for Computing Machinery), 2005. p. 633–634. Trabalho apresentado no PROCEEDINGS OF THE 28TH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 2005, [Salvador, Brasil].

LIU, Z.; LIU, H. Several efficient gradient methods with approximate optimal stepsizes for large scale unconstrained optimization. **Journal of Computational and Applied Mathematics**, Elsevier, v. 328, p. 400–413, 2018.

LÓPEZ, J.; SUYKENS, J. A. First and second order smo algorithms for ls-svm classifiers. **Neural Processing Letters**, Springer, v. 33, p. 31–44, 2011.

LUND, B. D.; AGBAJI, D.; MANNURU, N. R. Perceptions of the fourth industrial revolution and ai's impact on society. **Perspectives on Global Development and Technology**, Brill, v. 23, n. 5-6, p. 385–406, 2024.

MADALA, H. R. **Inductive learning algorithms for complex systems modeling**. [*S. l.*]: CRC press, 2019.

MALL, R.; SUYKENS, J. A. Very sparse lssvm reductions for large-scale data. **IEEE transactions on neural networks and learning systems**, IEEE, v. 26, n. 5, p. 1086–1097, 2015.

MARINHO, F. P.; NETO, A. R. d. R.; ROCHA, P. A. C. Previsao de irradiância solar de curto prazo utilizando modelo de envelopes para os preditores. *In:* **CONGRESSO BRASILEIRO DE AUTOMÁTICA (CBA)**. Fortaleza, Brasil: SBA (Sociedade Brasileira de Automática), 2022. v. 3, n. 1. Trabalho apresentado no CONGRESSO BRASILEIRO DE AUTOMÁTICA (CBA), 2022, [Fortaleza, Brasil].

MARINHO, F. P.; ROCHA, P. A.; NETO, A. R. R.; SANTOS, V. O. Dimensional reduction for solar irradiance forecasting problem using principal components analysis and turk-pentland strategy. *In:* **INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN)**. Yokohama, Japão: IEEE, 2024. p. 1–9. Trabalho apresentado no INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN), 2024, [Yokohama, Japão].

MARINHO, F. P.; ROCHA, P. A. C.; SILVA, M. E. V. da; LIMA, R. J. P.; NETO, J. P. de A. Resultados preliminares de previsão de irradiação solar de curto prazo através da combinação de processamento de imagens com algoritmos de aprendizagem de máquina. *In:* **ANAIS CONGRESSO BRASILEIRO DE ENERGIA SOLAR (CBENS)**. [*S. l.*: *s. n.*], 2020. Trabalho apresentado no CONGRESSO BRASILEIRO DE ENERGIA SOLAR (CBENS), 2020.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. **Journal of the society for Industrial and Applied Mathematics**, SIAM, v. 11, n. 2, p. 431–441, 1963.

MEHRKANOON, S.; ZELL, A.; SUYKENS, J. A. Scalable hybrid deep neural kernel networks. *In:* **ESANN**. [*S. l.*: *s. n.*], 2017. p. 26–28.

MINSKY, M.; PAPERT, S. An introduction to computational geometry. **Cambridge tiass., HIT**, v. 479, n. 480, p. 104, 1969.

MORÉ, J. J. The levenberg-marquardt algorithm: implementation and theory. *In:* **NUMERICAL ANALYSIS: PROCEEDINGS OF THE BIENNIAL CONFERENCE HELD AT DUNDEE, JUNE 28–JULY 1, 1977**. Berlin, Alemanha: Springer, 2006. p. 105–116. Trabalho apresentado no NUMERICAL ANALYSIS: PROCEEDINGS OF THE BIENNIAL CONFERENCE HELD AT DUNDEE, JUNE 28–JULY 1, 1977, 1977, [Berlin, Alemanha].

NETO, A. R. d. R.; BARRETO, G. A. d. A. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: a comparative analysis. **IEEE Latin America Transactions**, IEEE, v. 7, n. 4, p. 487–496, 2009.

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

NOCEDAL, J. Updating quasi-newton matrices with limited storage. **Mathematics of computation**, v. 35, n. 151, p. 773–782, 1980.

OLIVEIRA, S. A. F.; GOMES, J. P. P.; NETO, A. R. R. Sparse least-squares support vector machines via accelerated segmented test: a dual approach. **Neurocomputing**, Elsevier, v. 321, p. 308–320, 2018.

OORD, A. v. d.; DIELEMAN, S.; ZEN, H.; SIMONYAN, K.; VINYALS, O.; GRAVES, A.; KALCHBRENNER, N.; SENIOR, A.; KAVUKCUOGLU, K. Wavenet: a generative model for raw audio. **arXiv preprint arXiv:1609.03499**, 2016. Disponível em: https://arxiv.org/abs/1609.03499. Acesso em: 09/10/2025.

PARK, J.; AHN, H.; KIM, D.; PARK, E. Gnn-ir: Examining graph neural networks for influencer recommendations in social media marketing. **Journal of Retailing and Consumer Services**, Elsevier, v. 78, p. 103705, 2024.

PEDRO, H. T.; COIMBRA, C. F. Nearest-neighbor methodology for prediction of intra-hour global horizontal and direct normal irradiances. **Renewable Energy**, Elsevier, v. 80, p. 770–782, 2015.

PEDRO, H. T.; LARSON, D. P.; COIMBRA, C. F. A comprehensive dataset for the accelerated development and benchmarking of solar forecasting methods. **Journal of Renewable and Sustainable Energy**, AIP Publishing LLC, v. 11, n. 3, p. 036102, 2019.

PENROSE, R. A generalized inverse for matrices. *In:* **MATHEMATICAL PROCEEDINGS OF THE CAMBRIDGE PHILOSOPHICAL SOCIETY**. [*S. l.*]: Cambridge University Press, 1955. v. 51, n. 3, p. 406–413. Trabalho apresentado no MATHEMATICAL PROCEEDINGS OF THE CAMBRIDGE PHILOSOPHICAL SOCIETY, 1955.

PÉREZ, L. L.; PRUDENTE, L. A wolfe line search algorithm for vector optimization. **ACM Transactions on Mathematical Software (TOMS)**, ACM New York, NY, USA, v. 45, n. 4, p. 1–23, 2019.

PLATT, J. **Sequential minimal optimization: a fast algorithm for training support vector machines**. Redmond, WA, 1998. Disponível em: https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/. Acesso em: 18/07/2025.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

SHAHAM, T. R.; DEKEL, T.; MICHAELI, T. Singan: Learning a generative model from a single natural image. *In:* **IEEE/CVF INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV)**. Seul, Coreia do Sul: IEEE, 2019. p. 4569–4579. Trabalho apresentado no IEEE/CVF INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2019, [Seul, Coreia do Sul].

SHEWCHUK, J. R. **An introduction to the conjugate gradient method without the agonizing pain**. Carnegie Mellon University, 1994.

SILVA, D. A.; SILVA, J. P.; NETO, A. R. R. N. Novel approaches using evolutionary computation for sparse least square support vector machines. **Neurocomputing**, Elsevier, v. 168, p. 908–916, 2015.

SUYKENS, J. A.; VANDEWALLE, J. Least squares support vector machine classifiers. **Neural processing letters**, Springer, v. 9, p. 293–300, 1999.

SUYKENS, J. A. K.; LUKAS, L.; VANDEWALLE, J. Sparse least squares support vector machine classifiers. *In:* **ESANN**. [*S. l.*: *s. n.*], 2000. p. 37–42. Trabalho apresentado no ESANN, 2000.

TEAM, R. C. R language definition. **Vienna, Austria: R foundation for statistical computing**, v. 3, n. 1, p. 116, 2000.

TORRES-BARRÁN, A.; ALAÍZ, C. M.; DORRONSORO, J. R. Faster svm training via conjugate smo. **Pattern Recognition**, Elsevier, v. 111, p. 107644, 2021.

TREFETHEN, L. N.; BAU, D. **Numerical linear algebra**. [*S. l.*]: SIAM, 2022.

TSAI, C.-W.; LAI, C.-F.; CHAO, H.-C.; VASILAKOS, A. V. Big data analytics: a survey. **Journal of Big data**, Springer, v. 2, p. 1–32, 2015.

TURING, A. M. Computing machinery and intelligence. *In:* **Parsing the Turing test: philosophical and methodological issues in the quest for the thinking computer**. [*S. l.*]: Springer, 2007. p. 23–65.

WANG, L.; CAO, M.; XING, F.; YANG, Y. The new spectral conjugate gradient method for large-scale unconstrained optimisation. **Journal of Inequalities and Applications**, Springer, v. 2020, p. 1–11, 2020.

WATKINS, D. S. **Fundamentals of matrix computations**. [*S. l.*]: John Wiley & Sons, 2004.

WRIGHT, S.; NOCEDAL, J. *et al.* Numerical optimization. **Springer Science**, v. 35, n. 67-68, p. 7, 1999.

XIA, X.-L. Training sparse least squares support vector machines by the qr decomposition. **Neural Networks**, Elsevier, v. 106, p. 175–184, 2018.

YAN, X.; SARKAR, M.; LARTEY, B.; GEBRU, B.; HOMAIFAR, A.; KARIMODDINI, A.; TUNSTEL, E. An online learning framework for sensor fault diagnosis analysis in autonomous cars. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 24, n. 12, p. 14467–14479, 2023.

YANG, X.; LU, J.; ZHANG, G. Adaptive pruning algorithm for least squares support vector machine classifier. **Soft computing**, Springer, v. 14, p. 667–680, 2010.

YU, L.; LI, S.; LIU, S. Fast support vector machine training via three-term conjugate-like smo algorithm. **Pattern Recognition**, Elsevier, v. 139, p. 109478, 2023.

YU, L.; MA, X.; LI, S. A fast conjugate functional gain sequential minimal optimization training algorithm for ls-svm model. **Neural Computing and Applications**, Springer, v. 35, n. 8, p. 6095–6113, 2023.

ZENG, X.; CHEN, X.-w. Smo-based pruning methods for sparse least squares support vector machines. **IEEE transactions on Neural Networks**, IEEE, v. 16, n. 6, p. 1541–1546, 2005.

ZHANG, L. An improved wei–yao–liu nonlinear conjugate gradient method for optimization computation. **Applied Mathematics and computation**, Elsevier, v. 215, n. 6, p. 2269–2274, 2009.

ZHANG, Z. Improved adam optimizer for deep neural networks. *In:* **IEEE/ACM INTERNATIONAL SYMPOSIUM ON QUALITY OF SERVICE (IWQoS)**. Banff, Alberta, Canada: IEEE, 2018. p. 1–2. Trabalho apresentado no IEEE/ACM INTERNATIONAL SYMPOSIUM ON QUALITY OF SERVICE (IWQoS), 2018, [Banff, Alberta, Canada].

ZHOU, S. Sparse lssvm in primal using cholesky factorization for large-scale problems. **IEEE transactions on neural networks and learning systems**, IEEE, v. 27, n. 4, p. 783–795, 2015.

ZHOU, S.; LIU, M. A new sparse lssvm method based the revised lars. *In:* **INTERNATIONAL CONFERENCE ON MACHINE VISION AND INFORMATION TECHNOLOGY (CMVIT)**. Cidade de Cingapura: IEEE, 2017. p. 46–51. Trabalho apresentado no INTERNATIONAL CONFERENCE ON MACHINE VISION AND INFORMATION TECHNOLOGY (CMVIT), 2017, [Cidade de Cingapura].

ZOUTENDIJK, G. Nonlinear programming, computational methods. **Integer and nonlinear programming**, North-Holland, p. 37–86, 1970.