



UNIVERSIDADE FEDERAL DO MARANHÃO
UNIVERSIDADE FEDERAL DO PIAUÍ
Doutorado em Ciência da Computação Associação
UFMA/UFPI

Jônatas Oliveira Lima da Silva

**Um novo framework para treinamento de Support Vector
Machines (SVM) e suas aplicações**

Orientador: Prof. Dr. Paulo Sérgio Marques dos Santos

Teresina - PI
Julho, 2024

Jônatas Oliveira Lima da Silva

**Um novo framework para treinamento de Support Vector
Machines (SVM) e suas aplicações**

TESE DE DOUTORADO

Tese apresentada como requisito parcial
para obtenção do título de Doutor em Ciência
da Computação, ao Doutorado em Ciência
da Computação, Associação UFMA/UFPI.

Orientador: Prof. Dr. Paulo Sérgio Marques dos Santos

Teresina - PI
Julho, 2024

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Sistema de Bibliotecas UFPI - SIBi/UFPI
Biblioteca Setorial do CCN

S586n Silva, Jônatas Oliveira Lima da.
Um novo framework pra treinamento de Support Vector
Machines (SVM) e suas aplicações / Jônatas Oliveira Lima da
Silva. -- 2024.
88 f. : color.

Tese (Doutorado) - Universidade Federal do Piauí. Centro
de Ciências da Natureza. Programa de Pós-Graduação em
Ciência da Computação, Teresina, 2024.
“Orientador: Prof. Dr. Paulo Sérgio Marques dos Santos.”

1. Support Vector Machines(SVM). 2. Redes Neurais
Convolucionais. 3. Lesão cutânea (câncer de pele). 4. Problema
da mochila quadrática separável. I. Santos, Paulo Sérgio
Marques dos. II. Título.

CDD 006.3

Bibliotecária: Caryne Maria da Silva Gomes - CRB3/1461

Jônatas Oliveira Lima da Silva

Um novo framework para treinamento de Support Vector Machines (SVM) e suas aplicações

A presente Tese de Doutorado foi avaliada por banca examinadora composta pelos seguintes membros:

Prof. Dr. Paulo Sérgio Marques dos Santos

Orientador

Universidade Federal do Delta do Parnaíba

Prof. Dr. Luiz Carlos Matioli

Universidade Federal do Paraná

Prof. Dr. Roberto Cristovão Mesquita Silva

Universidade Federal do Amazonas

Prof. Dr. Andre Macedo Santana

Universidade Federal do Piauí

Prof. Dr. Ricardo de Andrade Lira Rabelo

Universidade Federal do Piauí

Prof. Dr. João Carlos de Oliveira Souza

Universidade Federal do Piauí

Certificamos que esta é a versão original e final da Tese de Doutorado que foi julgada adequada para obtenção do título de Doutor em Ciência da Computação.



Documento assinado digitalmente

PAULO SERGIO MARQUES DOS SANTOS

Data: 08/01/2025 16:19:32-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Paulo Sérgio Marques dos Santos

Orientador



Documento assinado digitalmente

IVAN SARAIVA SILVA

Data: 08/01/2025 19:22:21-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Ivan Saraiva Silva

Coordenador

Teresina - PI, 31 de Julho de 2024

*Dedico este trabalho, primeiramente, a Deus,
que me deu força e disciplina para chegar até aqui
e aos meus pais, por serem um exemplo vivo
ao qual pude me espelhar.*

Agradecimentos

Agadeço primeiramente a Deus, por ter me tirado de um grande abismo no ano de 2017, me dando inspiração de ingressar na área acadêmica em 2018 quando iniciei o mestrado, e por ter me guiado e levado tão longe até o presente momento.

A minha esposa, Amanda Farias da Silva, que a conheci já nos últimos anos do doutorado, porém foi peça fundamental nesse período em que mais me foi exigido durante o programa. Nos momentos de maior estresse, é ela que consegue me tirar do trabalho, respirar um ar puro, ter novas idéias e resgatar a paz que as vezes perdemos em meio a tantas tarefas.

Ao meu pai, José Lima da Silva, que me ensinou por meio de suas atitudes e exemplos como conquistar coisas grandes através da determinação, e juntamente com minha mãe, Maria de Fátima Oliveira Soares, me deram a melhor educação que eu poderia ter. Educação essa que envolve caráter e respeito pela família e pelo próximo.

Ao meu orientador Professor Doutor Paulo Sergio Marques dos Santos, pela paciência que teve comigo desde o mestrado. Aprendi com ele não apenas como fazer pesquisa, mas os valores e contribuições da ciência para a sociedade. Com ele aprendi que tudo é possível com esforço. Uma frase que escutei sair dele nunca esqueci foi: "Se alguém conseguiu, a gente também consegue".

Aos professores e parceiros de pesquisa que me acompanham desde o mestrado, tenho um enorme carinho por todos, assim como os presentes professores membros desta banca examinadora.

À UFPI e UFMA, por toda a infraestrutura e auxílio que me foi concedido, e juntamente com o DCCMAPI e os demais professores e funcionários me proporcionaram todo o conhecimento e base necessária para chegar até aqui.

"Não se amolde ao padrão deste mundo, mas transforme-se."

(Romanos 12:1)

Resumo

Apresentamos um framework unificado para resolver problemas de treinamento não lineares de Support Vector Machines (SVM). O framework é baseado em uma aproximação da função objetivo, tornando o problema separável, e utiliza métodos de busca por um zero de uma função de baixo custo computacional para resolver os subproblemas resultantes. Devido à diagonalização da função objetivo no primeiro estágio do framework, nomeamos o novo solucionador de DiagSVM. Para testar o desempenho do DiagSVM, realizamos experimentos numéricos preliminares com conjuntos de dados de benchmark. A partir dos resultados, escolhemos a melhor combinação utilizada no framework para resolver o problema de Classificação de Lesões Cutâneas (CLC). Dado que o melanoma (câncer de pele) é a doença mais perigosa e mortal que afeta a pele, a aplicação do DiagSVM pode ser integrada em vários sistemas de diagnóstico assistido por computador (CAD) para ajudar na detecção do câncer de pele e reduzir significativamente a morbidade e mortalidade associadas a essa doença. Abordagens baseadas em aprendizado de máquina (ML) e aprendizado profundo (DL) têm sido amplamente utilizadas para desenvolver sistemas robustos de classificação de lesões cutâneas. Para o problema CLC, três redes neurais convolucionais (CNN) pré-treinadas – Xception, InceptionResNetV2 e DenseNet201 – foram empregadas como extratores de características, e suas dimensões foram reduzidas usando Análise de Componentes Principais (PCA), Análise de Componentes Principais de Kernel (KPCA) e Análise de Componentes Independentes (ICA). Para validar a proposta, comparamos o DiagSVM com o solucionador popular na literatura, o Libsvm.

Palavras-chave: Support Vector Machines (SVM), Aproximação quadrática separável, Problema da mochila quadrática separável, Métodos de busca por um zero de uma função, Redes Neurais Convolucionais (CNN).

Abstract

We present a unified framework for solving nonlinear Support Vector Machines (SVM) training problems. The structure is based on an objective function approximation so that the Problem becomes separable, with low computational cost root-finding methods to solve the resulting subproblems. Due to the diagonalization of the objective function in the first stage of the framework, we named the new SVM solver DiagSVM. To test the performance of DiagSVM, we report preliminary numerical experiments with benchmark datasets. Based on the results, we chose the best combination used in the framework to solve the Skin Lesions Classification (CLC) problem. Since melanoma (skin cancer) is the most dangerous and deadly disease affecting the skin, the DiagSVM application can be integrated into various computer-aided diagnosis (CAD) systems to help them detect skin cancer and significantly reduce associated morbidity and mortality. with this disease. Machine learning (ML) and deep learning (DL)-based approaches have been widely used to develop robust skin lesion classification systems. For the CLC problem, three pre-trained convolutional neural networks (CNN), Xception, InceptionResNetV2 and DenseNet201, were employed as feature extractors and their dimension was reduced using Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA) and Independent Component Analysis (ICA). To validate the proposal, we compared DiagSVM with the very popular solver in the literature, Libsvm.

Keywords: Support Vector Machines (SVM), Separable quadratic approximation, Separable quadratic knapsack problem, Root-finding methods, Convolutional Neural Networks (CNN).

Lista de ilustrações

Figura 1 – SVM de margem Rígida	25
Figura 2 – SVM de margem suave com destaque nas variáveis de folga	29
Figura 3 – (a) Base de dados não linear; (b) Hiperplano separador no espaço de características (WANG et al., 2024)	32
Figura 4 – Diagrama de blocos do DiagSVM	36
Figura 5 – Função <i>Softplus</i>	52
Figura 6 – Derivada da função <i>softplus</i>	52
Figura 7 – Função polinomial	53
Figura 8 – Derivada da função polinomial	54
Figura 9 – Função Elliot	55
Figura 10 – Número 0	63
Figura 11 – Números entre 1 e 9	63
Figura 12 – Gráfico de barra com a soma das acurácias obtidas em todos os algoritmos e em todos os conjuntos de dados de benchmark.	67
Figura 13 – Diagrama de blocos para classificação de lesões cutâneas	68
Figura 14 – Aumento de dados utilizando Horizontal Flip	70
Figura 15 – Representação da acurácia, sensibilidade e especificidade dos algoritmos em cada modelo aplicado a base de dados ISIC2017	74
Figura 16 – Representação da acurácias, sensibilidade e especificidade dos melhores algoritmos em cada modelo aplicado a base de dados PH2.	77

Lista de tabelas

Tabela 1 – Algumas funções <i>kernel</i>	35
Tabela 2 – Métricas de performance	60
Tabela 3 – Conjunto de dados	62
Tabela 4 – Significado das siglas dos algoritmos	64
Tabela 5 – Resultados da aproximação quasi-Cauchy com ponto fixo e método secante	65
Tabela 6 – Resultados da aproximação quasi-Cauchy com regula falsi e método de bissecção	65
Tabela 7 – Resultados da aproximação esférica com ponto fixo e método secante	65
Tabela 8 – Resultados da aproximação esférica com regula falsi e método de bissecção	66
Tabela 9 – Resultados do Libsvm	66
Tabela 10 – Descrição das classes usadas em problemas de classificação binária para o conjunto de dados ISIC2017 antes da etapa de aumento (augmentation).	69
Tabela 11 – Descrição das classes usadas nos problemas de classificação binária para o conjunto de dados PH2 antes da etapa de aumento (augmentation).	69
Tabela 12 – Descrição das classes usadas em problemas de classificação binária para o conjunto de dados ISIC2017 e PH2 após a etapa de aumento.	71
Tabela 13 – Resultados da classificação ISIC2017 usando Xception como extrator de características.	73
Tabela 14 – Resultados da classificação ISIC2017 usando InceptionResNetV2 como extrator de características.	74
Tabela 15 – Resultados da classificação ISIC2017 usando DenseNet201 como extrator de características.	75
Tabela 16 – Resultados da classificação PH2 usando Xception como extrator de características.	76
Tabela 17 – Resultados da classificação PH2 usando InceptionResNetV2 como extrator de características.	77
Tabela 18 – Resultados da classificação PH2 usando DenseNet201 como extrator de recursos.	78
Tabela 19 – Produções científicas em relação ao método proposto para classificação de lesões cutâneas.	81
Tabela 20 – Produções científicas em outras aplicações relacionadas a otimização matemática	81

Lista de abreviaturas e siglas

SVM	<i>Support Vector Machines</i>
Sp	<i>Aproximação esférica</i>
Qc	<i>Aproximação Quasi-Cauchy</i>
fp	<i>Método do ponto fixo</i>
sc	<i>Método secante</i>
rf	<i>Método regula falsi</i>
bs	<i>Método bisseção</i>
LIBSVM	<i>Library for SVM</i>

Sumário

1	INTRODUÇÃO	14
1.1	Objetivo Geral	16
1.2	Objetivos Específicos	16
1.3	Organização do Trabalho	17
2	TRABALHOS RELACIONADOS	18
2.1	Métodos de otimização aplicados ao SVM	19
2.2	Outros métodos de otimização	20
2.3	SVM Aplicado em Técnicas de Deep Learning: SVM e CNN	21
2.4	Considerações Finais do Capítulo	23
3	FUNDAMENTAÇÃO TEÓRICA	24
3.1	SVM Linearmente Separável (Margem Rígida)	25
3.2	SVM Não Linearmente Separável (Margem suave)	28
3.3	Funções Kernel	31
3.4	Considerações Finais do Capítulo	35
4	MÉTODO PROPOSTO	36
4.1	Primeiro Estágio do framework: Aproximações Quadráticas Separáveis	37
4.1.1	Aproximação Quadrática Esférica	37
4.1.2	Aproximação Quasi-Cauchy	40
4.1.3	Adaptando a Aproximação Quadrática Separável ao Framework	42
4.1.4	Outras Opções para o Primeiro Estágio do Framework	43
4.1.4.1	Aproximação Quasi-Newton	43
4.1.4.2	Técnicas de Região de Confiança e Conservadorismo	44
4.2	Segundo Estágio do Framework: Métodos de Busca por um Zero de uma Função	46
4.2.1	Método do Ponto Fixo	48
4.2.2	Método do Ponto Fixo Acelerado	49
4.2.2.1	Aceleração de Anderson	49
4.2.2.2	Aceleração de Aitken	50
4.2.3	Métodos baseados em aproximação da função <i>max</i>	50
4.2.3.1	Aproximação <i>softplus</i>	51
4.2.3.2	Aproximação polinomial	53
4.2.3.3	Aplicando aproximação da função <i>max</i> no problema PMQS	53

4.2.3.4	Extensões para aproximação da função max	55
4.2.4	Revisão dos métodos utilizados no segundo estágio do framework	55
4.3	Estrutura do Algoritmo Proposto	56
4.4	Implementação e Detalhes Técnicos	57
4.4.1	Complexidade do Algoritmo Proposto	58
4.5	Considerações Finais do Capítulo	58
5	EXPERIMENTOS NUMÉRICOS	60
5.1	Problemas de benchmark	61
5.1.1	Descrição do conjunto de dados de benchmark	61
5.1.2	Resultados numéricos para os problemas de benchmark	63
5.2	Problema de Classificação de Lesões Cutâneas (CLC)	67
5.2.1	Formulação do Problema CLC	67
5.2.1.1	Conjunto de dados de imagens dermatoscópicas	68
5.2.1.2	Aumento dos dados (augmentation)	70
5.2.1.3	Extração de características	70
5.2.1.4	Engenharia de recursos	71
5.2.1.5	Classificação	72
5.2.2	Resultados para a base de dados ISIC2017	72
5.2.3	Resultado para a base de dados PH2	74
5.3	Considerações Finais do Capítulo	77
6	CONCLUSÃO	79
6.1	Contribuições	79
6.2	Trabalhos Futuros	80
6.3	Produções Científicas	81
	REFERÊNCIAS	82

1 Introdução

Entre os diferentes métodos de última geração para resolver problemas de aprendizado de máquina está o *Support Vector Machine* (SVM). Ele combina excelente desempenho com uma base matemática sólida, conforme demonstrado por [Boser, Guyon e Vapnik \(1992\)](#) e posteriormente por [Cortes e Vapnik \(1995\)](#).

Atualmente, a comunidade de aprendizado de máquina enfrenta novos desafios na aplicação do SVM em cenários do mundo real, como a variedade, o volume, a velocidade e a veracidade dos dados. A quantidade de dados (e sua diversidade em termos de qualidade) gerada diariamente cresce de forma tremenda na maioria dos domínios científicos e de engenharia, incluindo áreas como finanças, imagem médica, categorização de textos e biologia computacional. Embora isso pareça vantajoso à primeira vista - mais dados podem significar uma maior chance de extrair informações valiosas e revelar conhecimento útil - o treinamento de SVM com base de dados aumentando a cada ano um problema fundamental devido ao alto tempo computacional e à complexidade de memória exigidos em tais aplicações ([LIU et al., 2017](#); [QIU et al., 2016](#)).

Ainda assim, o SVM é um dos algoritmos de aprendizado de máquina mais usado para diversos problemas de aprendizado, como diagnóstico de doenças por imagem, filtragem de spam, classificação de texto, análise de manuscritos, reconhecimento de rosto e objeto, entre muitos outros, conforme mostrado por [Cervantes et al. \(2020\)](#). Em [Bhavsar e Panchal \(2012\)](#), destacam duas características principais do SVM: a teoria da generalização, que leva a uma forma de escolher uma hipótese baseada em princípios, e as funções Kernel, que introduzem não linearidade no espaço de hipóteses sem exigir explicitamente um algoritmo não linear. Ambos os conceitos (generalização e funções Kernel) serão abordados mais a fundo nas próximas seções.

O SVM é um classificador binário supervisionado que resolve um problema de programação quadrático restrito (PQ) para encontrar um hiperplano que separe as classes. O problema de otimização presente no SVM pode se apresentar de duas formas: primal e dual. Desde a sua criação, pesquisadores propuseram diferentes reformulações e continuam a evoluir com novos solucionadores e estratégias para resolvê-lo. De acordo com [Cervantes et al. \(2020\)](#), a principal razão para múltiplas implementações é que o tempo computacional depende principalmente das heurísticas usadas para dividir o problema em pequenos fragmentos. A complexidade computacional dos SVMs é quase cúbica, de modo que, em grandes conjuntos de dados, o tempo de treinamento se torna enorme, sendo crucial utilizar algum algoritmo para mitigar esse desafio.

Algumas implementações bastante populares na literatura usam uma ou mais abordagens para reduzir o tempo de treinamento do SVM: *SVMLight* (JOACHIMS, 1999) é aplicado na forma dual do SVM. Ele usa um algoritmo de decomposição juntamente com seleção do conjunto de trabalho (elementos escolhidos para compor o subproblema decomposto) para melhorar o tempo de treinamento; *Pegasos* (SHALEV-SHWARTZ; SINGER; SREBRO, 2007) resolve a formulação primal do SVM e é essencialmente um algoritmo de otimização de subgradiente estocástico; O algoritmo *LIBSVM* (CHANG; LIN, 2011) é baseado no algoritmo *Sequential Minimal Optimization* (SMO) e é aplicado na formulação dual do SVM. *LIBSVM* usa um algoritmo de seleção do conjunto de trabalho mais avançado fazendo uso de heurísticas; Suykens e Vandewalle (1999) reformularam o problema PQ presente no treinamento do SVM em um sistema de equações lineares e propuseram o Least Squares Support Vector Machine (LSSVM) para resolvê-lo, obtendo assim uma fórmula fechada para o treinamento do SVM. Outra abordagem, mais semelhante à proposta apresentada nesse trabalho, consiste em resolver o problema PQ presente na forma dual do SVM, aplicando o método do gradiente projetado (BIRGIN; MARTÍNEZ; RAYDAN, 2014) e, em seguida, utilizando métodos de busca por um zero de uma função para resolver os subproblemas gerados (SERAFINI; ZANGHIRATI; ZANNI, 2005; DAI; FLETCHER, 2006; COMINETTI; MASCARENHAS; SILVA, 2014).

Neste contexto, esta tese introduz o framework DiagSVM, um novo solucionador de SVM desenvolvido para enfrentar os desafios computacionais associados ao treinamento de SVMs. O framework DiagSVM é composto por duas etapas principais. Na primeira, a função objetivo é aproximada de forma a tornar o problema separável, utilizando métodos como a aproximação esférica (SNYMAN; HAY, 2001) e a aproximação quasi-Cauchy (ZHU; NAZARETH; WOLKOWICZ, 1997). Na segunda etapa, são aplicados métodos de busca por um zero de uma função para resolver os subproblemas resultantes. Esses métodos incluem estratégias clássicas como bisseção, regra falsi (MÜNNICH; SACHS; WAGNER, 2012), e métodos do ponto fixo (ALVES et al., 2023), além de novas abordagens desenvolvidas especificamente para o DiagSVM.

Para validar a eficácia do DiagSVM, foram realizados experimentos numéricos com conjuntos de dados de benchmark amplamente utilizados na literatura, como MNIST, heart, australian, fourclass, ionosphere, sonar e splice. Esses experimentos demonstraram que o DiagSVM não só reduz significativamente o tempo de treinamento, mas também mantém a precisão e a capacidade de generalização competitivas em relação aos métodos de SVM tradicionais, como o Libsvm. Em particular, os resultados mostraram que o DiagSVM é capaz de treinar conjuntos de dados de tamanho médio, mantendo as métricas de desempenho comparáveis às soluções existentes.

Além da validação com dados de benchmark, o DiagSVM foi aplicado ao problema de classificação de lesões cutâneas (CLC), um problema crítico na detecção precoce de

melanoma, que é um dos tipos mais mortais de câncer de pele. Utilizando conjuntos de dados dermoscópicos, como ISIC2017 e PH2, o DiagSVM foi combinado com técnicas avançadas de extração de características baseadas em redes neurais convolucionais (CNN) pré-treinadas, como Xception, InceptionResNetV2 e DenseNet201. Os resultados indicaram que a redução de dimensionalidade das características extraídas, utilizando PCA, KPCA e ICA, melhora o desempenho do SVM aplicado a CLC.

A análise de imagens médicas, particularmente a detecção de câncer de pele, é uma área onde a eficiência do treinamento de SVMs é crucial. O câncer de pele, incluindo o melanoma, é uma das formas mais comuns e perigosas de câncer, tornando a detecção precoce e precisa essencial para o tratamento eficaz e a redução da mortalidade. Tradicionalmente, o diagnóstico de lesões cutâneas depende da avaliação visual por dermatologistas, um processo sujeito a erros humanos e variações. A incorporação de técnicas de aprendizado de máquina em sistemas de Diagnóstico Assistido por Computador (Computer-Aided Diagnosis, CAD) pode potencialmente aumentar a precisão e a consistência dos diagnósticos, melhorando significativamente os resultados clínicos.

Portanto, este trabalho não só propõe uma solução inovadora para o problema de treinamento de SVM em larga escala, como também demonstra sua aplicabilidade e eficácia em um problema real e de grande relevância na área médica. As contribuições deste estudo abrem caminho para futuras pesquisas em métodos de otimização separáveis e em novas aplicações de SVM em áreas críticas da saúde.

1.1 Objetivo Geral

Este trabalho visa desenvolver e validar um novo framework para resolver o problema de classificação utilizando SVM, denominado DiagSVM, que visa reduzir o custo computacional do processo de treinamento, mantendo a precisão e a capacidade de generalização do modelo. Além disso, propomos validar a eficácia do DiagSVM em conjuntos de dados de benchmark e em aplicações práticas, especificamente na classificação de lesões cutâneas.

1.2 Objetivos Específicos

1. Desenvolver o Framework DiagSVM: Implementar a aproximação esférica e a aproximação quasi-Cauchy para a função objetivo dos SVM, transformando o problema original em subproblemas separáveis.
2. Implementar Métodos de Busca por um Zero de uma Função: Aplicar e testar métodos clássicos de busca por um zero de uma função, como bisseção, regula falsi e métodos do ponto fixo, para solucionar os subproblemas gerados pelo DiagSVM.

3. Validar o Framework com Dados de Benchmark: Realizar experimentos numéricos utilizando conjuntos de dados como MNIST, heart, australian, fourclass, ionosphere, sonar e splice para comparar o desempenho do DiagSVM com métodos tradicionais de SVM.
4. Aplicar o DiagSVM na Classificação de Lesões Cutâneas: Utilizar conjuntos de dados dermoscópicos, como ISIC2017 e PH2, e combinar o DiagSVM com técnicas de extração de características baseadas em redes neurais convolucionais pré-treinadas para avaliar o desempenho do framework na detecção de melanoma.

1.3 Organização do Trabalho

Este trabalho está estruturado da seguinte forma: O Capítulo 2 descreve trabalhos relacionados ao tema, destacando aqueles das quais extraímos os conhecimentos necessários para elaboração desta tese. O Capítulo 3 trata da fundamentação teórica das técnicas utilizadas. Aprofundamos conceitos relacionados ao framework proposto e ao SVM, além dos problemas de otimização que propomos resolver. O Capítulo 4 apresenta a metodologia proposta para este trabalho; são organizados os métodos que compõem o framework proposto e sua aplicação no treinamento do SVM. O Capítulo 5 exibe as tabelas e gráficos contendo os experimentos numéricos, assim como análise dos resultados obtidos. Por fim, o Capítulo 6 apresenta as considerações finais sobre os resultados, trabalhos desenvolvidos e trabalhos futuros.

2 Trabalhos Relacionados

Muito esforço já foi dedicado à implementação de um método eficiente para resolver o problema de treinamento do SVM em larga escala. O SVM, como o conhecemos hoje, foi primeiramente proposto em [Boser, Guyon e Vapnik \(1992\)](#) e, em seguida, estendido em [Cortes e Vapnik \(1995\)](#). Em muitas aplicações práticas do SVM, solucionadores padrão que consistem em armazenar a matriz Hessiana do problema PQ podem não ser eficientes ou, em alguns casos, podem ser proibitivos devido à exigência de uma grande quantidade de espaço na memória. Na literatura, tais problemas são comumente resolvidos em duas etapas: a primeira consiste em decompor o problema PQ do SVM em vários conjuntos de trabalho, que consistem em problemas PQ menores; na segunda etapa, cada conjunto de trabalho é resolvido utilizando um método de otimização aplicado ao problema PQ.

O foco deste trabalho será o desenvolvimento de um algoritmo aplicado ao treinamento do SVM para classificar conjuntos de dados com classes binárias e não-linearmente separáveis, ou seja, lidamos com o SVM não-linear. A capacidade do SVM de classificar dados não-lineares vem com o uso das funções Kernel. Mais detalhes a respeito das funções Kernel serão abordados no Capítulo 3. Em [Chauhan, Dahiya e Sharma \(2019\)](#) é realizado um levantamento das principais formulações desde a publicação de [Boser, Guyon e Vapnik \(1992\)](#) e [Cortes e Vapnik \(1995\)](#).

De acordo com [Piccialli e Sciandrone \(2022\)](#), pode-se descrever os seguintes fatores para determinar um algoritmo para SVM:

- Propriedades da formulação do problema, como a função objetivo ser convexa ou não, a função objetivo ser ou não suave, ser um problema restrito ou irrestrito ou até mesmo resolvido em sua formulação primal ou dual;
- Propriedades dos dados, relacionadas a um conjunto com mais amostras do que características ou vice-versa, um conjunto esparso, entre outras propriedades;
- O que se espera com o algoritmo, isto é, a acurácia é mais importante ou o tempo computacional é mais importante, etc.

Neste capítulo, apresentamos alguns dos principais algoritmos na literatura com o propósito de mitigar as dificuldades do problema de classificação utilizando a ideia do SVM. Em seguida, são apresentadas algumas referências que originaram a ideia do framework proposto, assim como aplicações do SVM no diagnóstico de doenças por imagens.

2.1 Métodos de otimização aplicados ao SVM

Existem vários algoritmos destinados a resolver o problema de SVM, seja ele na formulação primal ou dual. Para um problema de classificação binária e de acordo com a função de perda utilizada, pode-se citar os pacotes Liblinear (FAN et al., 2008) e Libsvm (CHANG; LIN, 2011). Além destes, vale ressaltar os seguintes métodos: Pegasos (SHALEV-SHWARTZ; SINGER; SREBRO, 2007), SVM^{light} (JOACHIMS, 1999), SVM^{perf} (JOACHIMS, 2006). Em Lin et al. (2009) e Lucidi et al. (2009) são estudados métodos de composição designados especificamente ao problema de SVM. Método do gradiente projetado aplicado a SVM foi desenvolvido em Serafini, Zanghirati e Zanni (2005), e uma versão paralela ainda é proposto em Zanni et al. (2006). Descreve-se abaixo um resumo das ideias dos principais métodos para um problema de classificação binária.

- LIBrary for large LINEAR classification (LIBLINEAR), proposto por Fan et al. (2008), é destinado ao problema de SVM tanto na forma primal quanto dual. Este pacote utiliza os algoritmos de Hsieh et al. (2008) ou Lin, Weng e Keerthi (2007), os quais descrevemos abaixo:
 - O método de coordenadas descendentes, proposto por Hsieh et al. (2008), é aplicado a uma formulação dual do SVM. Sua implementação é vantajosa devido ao problema dual possuir o Kernel linear, fazendo com que, para um conjunto de dados de alta dimensão e esparsos, atinja uma solução com uma tolerância ϵ de complexidade computacional $\log(1/\epsilon)$;
 - Trust RegiOn Newton (TRON), proposto por Lin, Weng e Keerthi (2007), emprega a abordagem região de confiança juntamente com Gradiente Conjugado pré-condicionado, utilizando uma fatoração de Cholesky incompleta, resolvendo a cada iteração um problema dual.
- LIBrary for SVM (LIBSVM), proposto por Chang e Lin (2011), utiliza o método de decomposição *Sequential Minimal Optimization* (SMO), proposto por Platt (1998). A ideia do SVM é resolver o problema PG do SVM, com funções Kernel linear ou não, decompondo-o em somente duas variáveis e fixando as demais na iteração. Com essa redução, o problema na iteração pode ser resolvido analiticamente. Para a escolha dessa duas variáveis usa-se heurística. O SMO trata-se portanto de um caso extremo de decomposição, logo, para um conjunto de amostras de alta dimensão, o Libsvm torna-se lento.
- SVM^{light}, proposto por Joachims (1999), resolve o problema dual do SVM, tanto com Kernel ou não. A ideia é resolver iterativamente o problema PQ, que possui restrição linear e de caixa, ao decompô-lo em um determinado conjunto de variáveis ativas, ao qual serão as variáveis otimizadas na iteração, fixando as demais variáveis. A

seleção das variáveis para compor este conjunto de trabalho é realizado de forma a auxiliar no progresso de otimização da função objetivo. Assim, resolve-se a cada iteração uma sequência de subproblemas PQ até um critério de parada ser satisfeito. No geral, a complexidade computacional dos algoritmos de decomposição é superlinear para um conjunto de n amostras. Por sua vez, uma desvantagem é que algoritmos de decomposição precisam de um longo tempo para treinar o modelo, implicando numa taxa de convergência lenta para o ótimo na função objetivo do problema primal.

- SVM Performance (SVM^{perf}), proposto por [Joachims \(2006\)](#), é baseado no algoritmo de planos de cortes a partir de uma formulação alternativa do SVM linear, mas que continua sendo equivalente. Tal formulação é adotada devido à forma esparsa que a mesma traz em comparação com formulações convencionais.
- Primal Estimated sub-GrAdient SOLver for SVM (PEGASOS), proposto por [Shalev-Shwartz, Singer e Srebro \(2007\)](#), é baseado na abordagem de sub-gradientes estocásticos aplicado a formulação primal do problema irrestrito, tanto para o caso linear como não-linear. No caso de Kernel linear, este método é indicado para um conjunto de dados de treino com alta dimensão devido sua complexidade computacional não depender diretamente da quantidade de amostras. Já o caso não linear depende da quantidade de amostras, o que o torna não indicado para um conjunto de dados de alta dimensão.
- Método do gradiente projetado aplicado ao problema PQ do SVM foi desenvolvido em [Serafini, Zanghirati e Zanni \(2005\)](#). A ideia é semelhante ao $\text{SVM}^{\text{light}}$, porém designado para resolver subproblemas maiores em cada iteração no método de decomposição. Experimentos mostraram que a técnica de decomposição juntamente com o método de gradiente projetado permitiram que o problema de otimização fosse resolvido em menos iterações.

2.2 Outros métodos de otimização

Com relação ao framework proposto, como já mencionado, não foi encontrado na literatura uma abordagem que combine aproximação diagonal com métodos de busca por um zero de uma função para resolver o problema de classificação de SVM. Então será apresentado abaixo algumas referências que motivaram o framework proposto.

A aproximação diagonal da Hessiana na qual a proposta desse trabalho se baseou, iniciou com os trabalhos de [Snyman e Hay \(2001\)](#), para resolver problemas de otimização irrestritos. A abordagem é considerada uma alternativa para os métodos baseados em gradiente conjugado por requerer menos armazenamento de variáveis e por

sua eficiência computacional. O método aplica a técnica do gradiente descendente em sucessivas aproximações quadráticas esféricas da função objetivo de tal forma que nenhuma busca linear (*line search*) seja necessária na resolução do problema de minimização. [Snyman e Hay \(2001\)](#) mostram ainda que o método é convergente quando aplicado a funções quadráticas positivas-definidas gerais e se sai bem com problemas mal condicionados. Em [Zhu, Nazareth e Wolkowicz \(1997\)](#), é apresentado ainda a aproximação Quasi-Cauchy, no qual é relacionada com a aproximação Quasi-Newton "fraca" de [\(DENNIS JR; WOLKOWICZ, 1993\)](#), onde uma restrição adicional substitui a matriz cheia por uma matriz diagonal. Além desses, existem outros trabalhos na literatura que podem ser estudados e analisados a possibilidade de ser aplicados no primeiro estágio do framework para aproximar uma matriz Hessiana cheia por uma matriz Hessiana diagonal, são eles: [Andrei \(2019\)](#), [Andrei \(2020\)](#), [Andrei \(2021\)](#), [Etman, Groenwold e Rooda \(2012\)](#) e suas referências.

O estudo de métodos de busca por um zero de uma função aplicado ao problema quadrático separável restrito se faz necessário pois consiste no segundo estágio do framework proposto. Alguns métodos existentes na literatura resolvem bem o problema, são eles: [Dai e Fletcher \(2006\)](#), [Cominetti, Mascarenhas e Silva \(2014\)](#), [Kiwiel \(2008\)](#) e [Münnich, Sachs e Wagner \(2012\)](#). Além destes, outras propostas de resolução do problema PMQS foram e estão sendo desenvolvidas ao longo da pesquisa do framework proposto, são elas: [Alves et al. \(2023\)](#) e [Alves, Silva e Santos \(2023\)](#).

2.3 SVM Aplicado em Técnicas de Deep Learning: SVM e CNN

A combinação de SVM com técnicas de deep learning, particularmente redes neurais convolucionais (CNN), tem mostrado resultados promissores, especialmente em tarefas de classificação de imagens. A integração desses métodos aproveita a capacidade das CNNs de extrair características relevantes de maneira automática e eficiente, enquanto os SVMs são utilizados para a classificação final dessas características extraídas. Conforme discutido em [Hasan et al. \(2023\)](#), as características extraídas de forma manual são muitas vezes desafiadoras devido ao aparecimento de vários ruídos nas imagens dermatoscópicas. Por outro lado, as abordagens baseadas em CNN fornecem excelentes resultados para CLC e aumentam as taxas de acertos do diagnóstico, ao mesmo tempo que são sistemas ponta a ponta. Muitos autores empregaram modelos CNN pré-treinados e os ajustaram (fining tuning) com os conjuntos de dados de lesões cutâneas ([HASAN et al., 2023](#)). A etapa de extração de características é realizada automaticamente pela CNN, o que pode ser eficaz no processo de classificação. Um modelo CNN geralmente compreende três camadas, denominadas convolução, pooling e camadas totalmente conectadas (fully connected layers) ([SCHMIDHUBER, 2015](#)). As

camadas de convolução e pooling desempenham uma função de extração de recursos, enquanto a camada totalmente conectada geralmente atua como um classificador. Embora a CNN possa classificar imagens, sua combinação com outros classificadores pode aumentar a eficiência da classificação, [Alizadeh e Mahloojifar \(2021\)](#). ([MAHBOD et al., 2019](#)) propôs um método computadorizado automático para classificação de lesões cutâneas, que emprega recursos profundos de CNNs. Eles usam três modelos profundos pré-treinados. Os recursos extraídos foram utilizados para treinar classificadores SVM e obtiveram uma taxa de precisão de 83,83%. A base de dados utilizada é o ISIC ([CODELLA et al., 2018](#)). [Yu et al. \(2017\)](#) apresentou uma estrutura de classificação híbrida para avaliação de imagens dermatoscópicas combinando CNN, vetor de Fisher (FV) e SVM linear.

Embora a CNN extraia automaticamente os recursos das imagens, a seleção dos recursos pode fornecer melhor desempenho e precisão aos sistemas. Portanto, selecionar o melhor subconjunto de recursos para classificação é essencial. Em [Khan et al. \(2019\)](#), os autores propuseram um sistema automatizado para CLC por meio da extração de recursos de rede neural profunda baseada em transferência de aprendizagem (DCNN) e seleção de recursos baseada em componente principal controlado por curtose (KcPCA). Eles usaram ResNet pré-treinado para extração de recursos e selecionaram os melhores recursos para treinar o modelo de classificação SVM. Em [Alizadeh e Mahloojifar \(2021\)](#), é proposto um sistema para detectar melanoma automaticamente usando CNNs e extração de características de textura de imagem. A dimensão de recursos também foi reduzida usando a análise de componentes principais de kernel (KPCA) para melhorar o desempenho da classificação. O método foi avaliado em ISIC2016, ISIC2019 e PH2.

A fase de classificação consiste em reconhecer e interpretar informações sobre lesões cutâneas com base nas características extraídas e selecionadas. O processo de classificação geralmente é realizado dividindo aleatoriamente as amostras de imagens disponíveis em conjuntos de treinamento e teste. A etapa de treinamento consiste no desenvolvimento de um modelo de classificação a ser empregado por um ou mais classificadores com base nas amostras do conjunto de treinamento. Cada amostra inclui características extraídas de uma imagem fornecida e seu valor de classe correspondente, nomeado como dados de entrada para o classificador. A etapa de teste mede as métricas do modelo aprendidas pela etapa de treinamento no conjunto de teste. [Hasan et al. \(2023\)](#) mostram que de 2011 a 2022, SVM e K-nearest Neighbours (KNN) foram os principais modelos de ML usados para CLC.

Um estudo notável de [Guo et al. \(2016\)](#) explorou a utilização de CNNs pré-treinadas como extratoras de características, seguidas pelo uso de SVMs para a classificação final. Essa metodologia mostrou-se eficaz ao combinar a capacidade de generalização das CNNs com o poder discriminativo dos SVMs, resultando em uma

melhoria substancial na performance do sistema. Especificamente, redes como AlexNet, VGG16 e ResNet foram usadas para extrair características de imagens, que então foram reduzidas em dimensão utilizando PCA antes de serem classificadas por SVMs. Os resultados mostraram que essa combinação supera métodos tradicionais que utilizam apenas CNNs ou SVMs isoladamente.

Outro exemplo significativo é o trabalho de [Tang \(2013\)](#), que aplicou CNNs e SVMs para a detecção de objetos em imagens, destacando que a combinação dessas técnicas pode lidar melhor com a variação e a complexidade das características visuais presentes em imagens reais. Tang demonstrou que a utilização de SVMs na camada de classificação final das CNNs melhora a capacidade de discriminação do modelo, resultando em maior precisão e menores taxas de erro.

O estudo de [Alquran et al. \(2017\)](#) sobre a detecção de melanoma utilizando SVMs em imagens dermoscópicas destaca a eficiência dos SVMs em identificar características malignas com alta precisão. Jafari et al. utilizaram uma abordagem baseada em texturas e características de cor extraídas das imagens, aplicando SVMs para a classificação final. Os resultados mostraram que os SVMs superaram outros métodos de classificação, especialmente em termos de sensibilidade e especificidade, crucial para diagnósticos médicos.

Em resumo, a literatura mostra que a integração de SVMs com técnicas de deep learning, como CNNs, e a aplicação de métodos de otimização na formulação dual dos SVMs, têm potencial significativo para avançar o estado da arte em classificação de imagens e diagnósticos médicos. A robustez e a alta precisão dos SVMs, combinadas com a capacidade de extração de características das CNNs, oferecem uma abordagem poderosa para resolver problemas complexos de classificação e diagnóstico.

2.4 Considerações Finais do Capítulo

Neste capítulo, foi apresentada uma revisão das produções científicas relevantes relacionadas aos métodos de otimização aplicados aos treinamento de SVM. Os métodos e os resultados obtidos nestes trabalhos foram brevemente apresentados. Algumas comparações foram feitas a respeito das diferenças entre as abordagens apresentadas, com o propósito de ligar as pesquisas realizadas com as que vem sendo produzidas pela comunidade científica em relação a esse tema. Além disso, foi apresentada uma seção dedicada às aplicações e contribuições do SVM na área da saúde assim como o desenvolvimento híbrido do SVM com CNN.

No próximo capítulo, serão apresentados conceitos teóricos fundamentais para o desenvolvimento deste trabalho.

3 Fundamentação Teórica

Este capítulo apresenta os principais conceitos necessários para entendimento do problema de treinamento e classificação do SVM. O SVM é uma técnica de aprendizado de máquina usada para classificação e análise de regressão. As técnicas de ML consistem em tirar conclusões gerais com base em exemplos particulares. A aprendizagem pode ser dividida em dois tipos principais: supervisionada e não supervisionada.

No aprendizado supervisionado, existe a figura de um agente externo, que apresenta o conhecimento do ambiente através de conjuntos de exemplos na forma: entrada, saída (CRISTIANINI; SHAW-TAYLOR et al., 2000). O algoritmo de aprendizado de máquina extrai a representação do conhecimento desses exemplos. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas anteriormente.

No aprendizado não supervisionado, não há agente externo presente, ou seja, não há exemplos rotulados. O algoritmo de aprendizado de máquina aprende a representar (ou agrupar) as entradas enviadas de acordo com uma medida de qualidade. Essas técnicas são usadas principalmente quando o objetivo é encontrar padrões ou tendências que ajudem a entender os dados (LORENA; CARVALHO, 2007).

O tipo de aprendizado abordado neste trabalho é supervisionado e será utilizado para análise de classificação. Nesse caso, um conjunto de dados de entrada é fornecido na forma de (x_i, y_i) , em que x_i possui m atributos, isto é, $x_i = (x_{i1}, \dots, x_{im})$ representa um elemento e y_i denota seu rótulo. A partir do conjunto de dados de entrada, um classificador, também chamado de modelo, deve ser produzido. Este classificador deve ser capaz de prever com precisão os rótulos de novos exemplos. Esse processo de gerar um classificador a partir de uma amostra de dados é chamado de treinamento. O classificador obtido também pode ser visto como uma função f , que recebe um dado x e fornece uma previsão y .

Para estimar a taxa de predições corretas ou incorretas obtidas por um classificador em novos dados, um conjunto de exemplos é, em geral, dividido em dois subconjuntos disjuntos: treinamento e teste. O subconjunto de treinamento é usado para gerar o classificador e o subconjunto de teste é usado para medir o grau de eficácia do classificador em prever a classe de novos dados.

O conjunto de entrada usado durante o estágio de treinamento também pode ser ou não linearmente separável. Baseado nessa separabilidade, o SVM possui algumas formulações que serão discutidas nas próximas seções.

3.1 SVM Linearmente Separável (Margem Rígida)

Neste primeiro momento, será tratado o método de classificação de máxima margem rígida, no qual é possível separar linearmente o conjunto de dados de entrada. Para dados que podem ser linearmente separáveis, é possível encontrar infinitos hiperplanos que conseguem separar os dados. Entretanto, o melhor hiperplano separador dos dados será aquele que cria uma maior margem entre determinados dados de entrada, conforme a Figura 1.

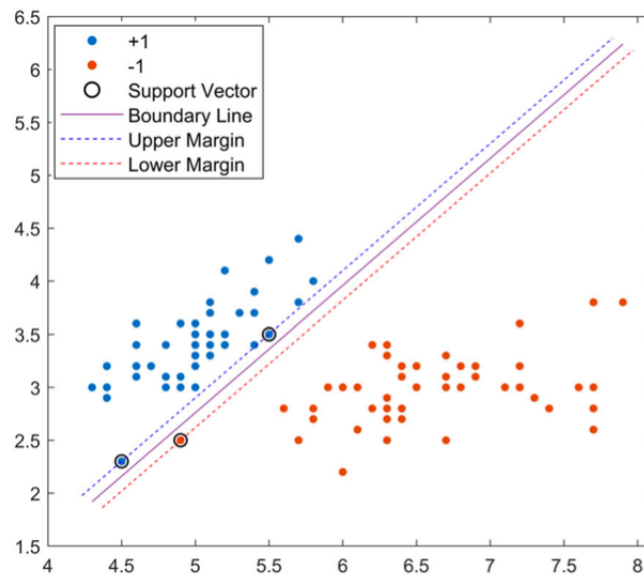


Figura 1 – SVM de margem Rígida

O classificador de margem rígida funciona apenas para dados linearmente separáveis, sendo assim, não pode ser utilizado em muitas situações do mundo real. No entanto, é o algoritmo mais fácil de entender, e forma o bloco de construção principal para os SVMs mais complexos. Ele exhibe os principais recursos que caracterizam esse tipo de máquina de aprendizado e, portanto, sua descrição é crucial para a compreensão dos sistemas mais avançados introduzidos posteriormente.

Considere T o conjunto de treinamento com n amostras $x_i \in X$ e seus respectivos rótulos $y_i \in Y$. $X \in \mathbb{R}^{n \times p}$ constitui o espaço de entrada, sendo que suas linhas representam as n amostras extraídas levando em consideração p características do problema e $Y = \{+1, -1\}$. Formalmente, podemos escrever T como:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), x_i \in \mathbb{R}^p, y_i = \{+1, -1\}\}. \quad (3.1)$$

Seguindo a teoria do classificador de margem máxima (CRISTIANINI; SHAWE-TAYLOR et al., 2000), consideramos dois conjuntos X_+ e X_- , formados pelos dados de entrada que pertencem às classes positivas e negativas, respectivamente, definidos como

$$X_+ = \{x_i \in X, y_i = 1\} \text{ e } X_- = \{x_i \in X, y_i = -1\}. \quad (3.2)$$

Com isso, o hiperplano separador H que dividirá os conjuntos X_+ e X_- , se dá por meio de um vetor $w \in \mathbb{R}^p$ não nulo e um escalar $b \in \mathbb{R}$ na forma da seguinte equação:

$$H = \{x \in \mathbb{R}^p, w^T x + b = 0\}. \quad (3.3)$$

Assim, tem-se que o hiperplano H dividirá qualquer espaço $\mathbb{R}^p$ em dois semi-espacos S_+ e S_- , no qual

$$S_+ = \{x \in \mathbb{R}^p, w^T x + b \geq 0\} \text{ e } S_- = \{x \in \mathbb{R}^p, w^T x + b \leq 0\}. \quad (3.4)$$

Nessas condições, os exemplos de cada classe poderão pertencer apenas ao lado positivo ($y = 1$) ou ao lado negativo ($y = -1$) do hiperplano. Assim, duas margens podem ser definidas para controlar a separabilidade dos dados de acordo com a equação abaixo:

$$w^T x + b \begin{cases} \geq 1 & \text{para } y_i = 1 \\ \leq -1 & \text{para } y_i = -1 \end{cases} \quad (3.5)$$

Como já mencionado, existem muitas margens que podem ser consideradas como bordas de cada classe. Portanto, para encontrar o melhor hiperplano, as distâncias d entre as margens devem ser maximizadas usando a seguinte equação (GHOLAMI; FAKHARI, 2017):

$$d(w, b; x) = \frac{|(w^T x + b - 1) - (w^T x + b + 1)|}{||w||} = \frac{2}{||w||} \quad (3.6)$$

Assim, maximizar a margem seria igual a minimizar o vetor dimensional w o qual pode ser escrito como $\frac{1}{2}w^T w$, e o seguinte problema de otimização convexa e com restrições lineares é definido (CRISTIANINI; SHAW-TAYLOR et al., 2000):

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}w^T w \\ \text{s.a.} \quad & y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, \dots, n, \end{aligned} \quad (3.7)$$

onde as restrições correspondem as margens das duas clases. O problema (3.7) possui solução global e única (x^*, b^*) , uma vez que a função objetivo deste problema é convexa e limitada inferiormente pelo conjunto de restrições (não vazio). Assim, definidos os valores de w^* e b^* , podemos classificar um novo $x \in \mathbb{R}^p$ ao empregar uma função de classificação $f : \mathbb{R}^n \rightarrow \mathbb{R}$ definida por

$$f(x) = \text{sinal}(w^{*T}x + b^*) = \begin{cases} \text{classe positiva} & , \text{ se } w^{*T}x + b^* > 0 \\ \text{classe negativa} & , \text{ se } w^{*T}x + b^* < 0 \end{cases} \quad (3.8)$$

O problema 3.7 representa a formulação primal para o classificador de máxima margem, porém a teoria de dualidade em otimização dispõe de uma forma mais simples de descobrir quais vetores seriam necessários para obter o hiperplano separador ótimo, além de contribuir na redução do custo computacional para encontrar o hiperplano neste caso. Logo, o multiplador de Lagrange α pode ser usado para gerar o novo problema (3.9):

$$L(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^n \alpha_i \{y_i[w^T x_i + b] - 1\} \quad (3.9)$$

sendo $\alpha \in \mathbb{R}^n$ o vetor de multiplicadores de Lagrange para as restrições do problema (3.7). A correspondente formulação dual é encontrada pela derivada parcial com respeito a w e b , impondo estacionaridade (MULLER et al., 2001):

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i x_i y_i \quad (3.10)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0. \quad (3.11)$$

Substituindo a equação (3.10) e (3.11) na equação (3.9), é obtido a equação geral para o classificador de máxima margem sujeito a duas restrições conforme a equação abaixo:

$$\begin{aligned} \text{maximizar } L(\alpha) = & -\frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j x_i^T x_j + \sum_{i=1}^n \alpha_i \\ \text{sujeito à} & \\ & \alpha_i \geq 0, \quad i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i y_i = 0. \end{aligned} \quad (3.12)$$

Desta forma, tem-se que o problema primal (3.7) possui solução única e global (w^*, b^*) e o problema dual (3.12) é um problema convexo com restrições lineares tendo α^* como solução. O exemplos no qual possuem $\alpha_i > 0$ são chamados vetores suporte (SV) e podem ser considerados como os dados mais informativos no conjunto de treinamento, pois são os únicos elementos participantes na determinação do equação do hiperplano separador (3.14) (os elementos circulados na Figura 1). w^* pode ser obtido a partir da (3.10). O parametro b^* é calculado a partir da equação (3.13) a seguir (MULLER et al.,

2001):

$$b^* = \frac{1}{n_{SV}} \sum_{x_i \in SV} \left(\frac{1}{y_j} - \sum_{x_i \in SV} \alpha_i^* y_i x_i^T x_j \right) \quad (3.13)$$

, onde n_{SV} denota o número de SV.

Como resultado final, o classificador $f(x)$ é apresentado na equação (3.14).

$$f(x) = \text{signal} \left(\sum_{i \in SV} y_i \alpha_i^* x_i^T x + b^* \right) = \begin{cases} \text{classe positiva} & , \text{ se } \sum_{i \in SV} y_i \alpha_i^* x_i^T x + b^* > 0 \\ \text{classe negativa} & , \text{ se } \sum_{i \in SV} y_i \alpha_i^* x_i^T x + b^* < 0 \end{cases} \quad (3.14)$$

Essa função linear representa o hiperplano que separa os dados com a máxima margem, considerando aquele com a melhor capacidade de generalização. Essa característica difere os SVMs de margem máxima da rede neural perceptron, onde o hiperplano obtido pode não ser aquele que apresenta a máxima margem entre as classes (LORENA; CARVALHO, 2007).

3.2 SVM Não Linearmente Separável (Margem suave)

Quando admite-se que o conjunto de dados utilizado não seja perfeitamente separável de forma linear, o modelo (3.7) deixa de ser útil para classificação, uma vez que seu conjunto de restrição formará uma região viável para resolução do problema. Geometricamente, para dados não separáveis linearmente significa que não é possível o problema (3.7) determinar qualquer hiperplano de forma a separar perfeitamente os dados de treinamento, pois ocorrem violações de restrições. Consequentemente, não se encontra solução para (3.7).

Mesmo assim, é possível encontrar um hiperplano que não separe perfeitamente os dados em duas classes, mas que forneça a melhor separação, de forma robusta, para a maioria dos dados. Isto é, permita que ocorra erros de classificação de algumas observações em troca de conseguir classificar a maioria dos dados. É nesse contexto que Cortes e Vapnik (1995) propõem uma flexibilização nas restrições do problema (3.7), através da adição de variáveis de folga ξ_i

$$\xi_i \geq 0, i = 1, 2, \dots, n,$$

que permitem violações ao conjunto de restrições do problema (3.7), reformulando-as para

$$1 - y_i(w^T x + b) \leq \xi_i \Leftrightarrow \left\{ \begin{array}{ll} w^T x_i + b \geq 1 - \xi_i & \forall x_i \in X_+ \\ w^T x_i + b \leq -1 + \xi_i & \forall x_i \in X_- \end{array} \right\}, \quad (3.15)$$

sendo que essas violações sejam as menores possíveis.

Observando-se (3.16), pode se tirar conclusões imediatas sobre a possibilidade de valores da variável de folga da nova definição do conjunto de restrições. Considere os três vetores $x_i, x_j \in X$, destacados na Figura 2, que pertencem a classe negativa, sendo que o hiperplano separador está representado pela linha sólida e suas respectivas margens estão ilustradas pelas linhas tracejadas.

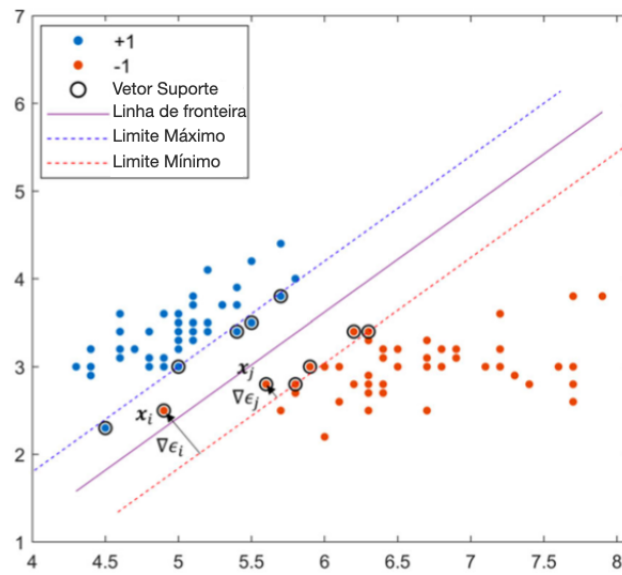


Figura 2 – SVM de margem suave com destaque nas variáveis de folga

Considerando que variáveis de folga ϵ_i e ϵ_j que permitam o mínimo de violações, como representado na Figura 2, pode-se concluir que:

- a variável ϵ_i assumirá um valor entre 0 e 1, pois o vetor x_i encontra-se no semiespaço negativo localizado entre a margem e o hiperplano separador;
- a variável ϵ_j assumirá um valor maior que 1, já que o vetor x_j encontra-se no semiespaço positivo. Assim, x_j estará do lado errado do hiperplano, lembrando que sua classe é negativa.

Então, quando alguma componente i do vetor ξ assume algum valor maior que zero, pode-se inferir que o valor da componente será exatamente a medida da violação ocorrida, isto é,

$$\xi_i = 1 - y_i(w^T x_i + b).$$

Para encontrar a menor violação possível ao passo de obter w^* e b^* otimizados, não é possível manter a mesma função objetivo como em (3.7), uma vez que a margem definida entre M_+ e M_- admitirá grandes valores arbitrários com as folgas ξ_i nas restrições. Isto fica visível quando, para quaisquer w_0 e b_0 que definem um hiperplano, pode-se encontrar $\xi_i \geq 0$ que satisfaçam (3.16), como por exemplo

$$\xi_i = \begin{cases} \max\{0, 1 - w_0^T x_i - b_0\} & \text{se } x_i \in X_+ \\ \max\{0, 1 + w_0^T x_i + b_0\} & \text{se } x_i \in X_- \end{cases}. \quad (3.16)$$

Logo, caso mantida a mesma função objetivo, isso implicará na possibilidade de admitir um hiperplano que classifique os dados de forma muito ruim às custas de introduzir variáveis de folga no conjunto de restrições. Assim, torna-se necessário penalizar as variáveis de folga na função objetivo afim de reduzir a quantidade de vetores com classificações equivocadas.

Logo, reformulando o problema (3.17) para um problema de classificação de máxima margem suave, obtemos o seguinte problema:

$$\begin{aligned} \min_{w,b} &= \frac{1}{2} w^T w + C \sum_{i=1}^n \xi_i \\ \text{s.t.} & \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, n. \end{aligned} \quad (3.17)$$

O parâmetro C na equação a cima é conhecida como "trade-off" ou parâmetro de regularização, adicionado para maximizar a margem e minimizar o erro de classificação (CRISTIANINI; SHAW-TAYLOR et al., 2000). Não é difícil ver que quanto menor for o valor do parâmetro de regularização C , mais próximo o problema (3.17) se torna do problema (3.7). É devido também ao parâmetro de regularização C que o problema (3.17) é conhecida como *C-Support Vector Classifier (C-SVC)*. Caberá a C permitir ou não a violação das restrições: um valor muito baixo de C permitirá que a margem do hiperplano separador seja grande, ocorrendo muitas violações; caso o valor seja alto, menor será a margem e a quantidade de violações. Então, assim como em Hastie et al. (2009), deve-se ter um equilíbrio na escolha de C , uma vez que:

- um valor baixo fará com que o ajuste do hiperplano aos dados seja influenciado pelas informações extras que os dados possam carregar (ruídos, por exemplo), ocorrendo uma menor variabilidade na definição do hiperplano nos conjuntos de treino possíveis (o que pode ser ruim na fase de testes de novas amostras);
- um valor alto fará com que as variáveis a serem ajustadas no modelo se adequem aos conjuntos de treino possíveis, variando mais a busca dos parâmetros deste ajuste com um viés mais baixo (o que pode ser ruim, caso seja ajustado um modelo com diferentes conjuntos de treino da mesma amostra X).

Como em [Gholami e Fakhari \(2017\)](#), os multiplicadores de Lagrange α e β podem ser usados para resolver o problema de otimização (3.17) obtendo o seguinte problema irrestrito:

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}w^T w + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i [w^T x_i + b] - 1 + \xi_i\} - \sum_{i=1}^n \beta_i \xi_i. \quad (3.18)$$

A solução ótima para o problema (3.9) pode ser alcançada satisfazendo as seguintes condições (KKT):

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i x_i y_i \quad (3.19)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \quad (3.20)$$

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow \alpha_i + \beta_i = C. \quad (3.21)$$

Substituindo as equações (3.19), (3.20) e (3.21) na equação (3.18), o seguinte problema dual SVM de margem suave é formulado ([GHOLAMI; FAKHARI, 2017](#)):

$$\begin{aligned} \text{maximizar } L(\alpha) &= -\frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j x_i^T x_j + \sum_{i=1}^n \alpha_i \\ \text{sujeito à} & \\ 0 \leq \alpha_i &\leq C, \quad i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i &= 0. \end{aligned} \quad (3.22)$$

3.3 Funções Kernel

Quando se refere ao SVM, não significa necessariamente o caso linear, pois ainda existem os chamados SVMs não lineares baseados na abordagem, chamada na literatura, de *Truque Kernel*. Como já mencionado, o propósito de usar um hiperplano ótimo é aumentar a capacidade de generalização da máquina, no entanto, mesmo encontrando o hiperplano ótimo, a máquina pode não ter uma boa capacidade de generalização devido à separabilidade não linear dos dados.

Para resolver esse problema, os dados de entrada são mapeados em um espaço de dimensão superior, também conhecido como espaço de características ou espaço de Hilbert, proposto por [Mercer \(1909\)](#). Em outras palavras, a ideia do *Truque do Kernel* será

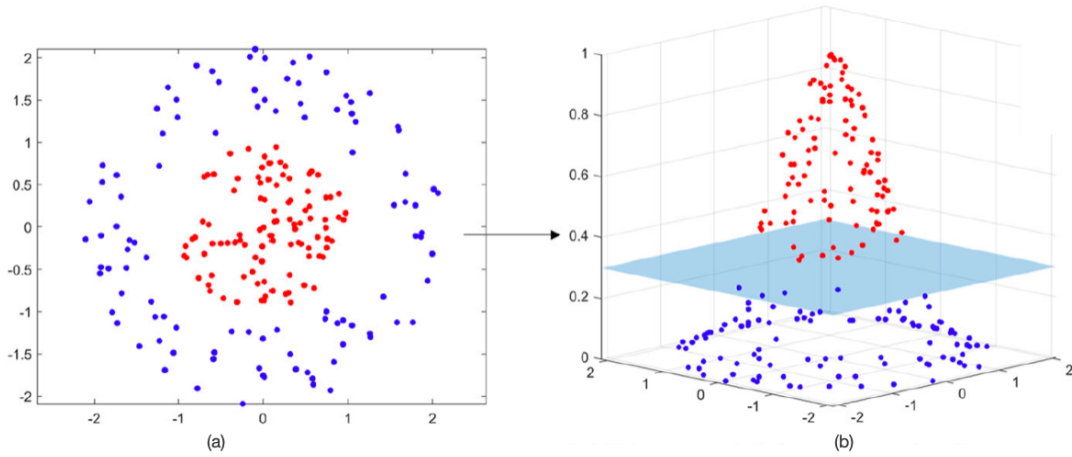


Figura 3 – (a) Base de dados não linear; (b) Hiperplano separador no espaço de características (WANG et al., 2024)

de aplicar transformações nos vetores do espaço original, levando-os para uma dimensão maior, denominado de espaço de características, e classificando-os nesta nova dimensão de acordo com um hiperplano separador a ser determinado. Mantendo essa teoria em mente, os dados ainda são não lineares no espaço de entrada, enquanto um hiperplano pode ser criado no espaço de características para separá-los. A Figura 3 exibe como o espaço de características pode ser usado para separar dados em uma dimensão superior (GHOLAMI; FAKHARI, 2017).

Assim, sabendo que as SVMs lidam com problemas não lineares mapeando o conjunto de dados de treinamento de seu espaço original, referido como entradas, para um novo espaço, chamado espaço de características, definimos $\phi : X \rightarrow \Omega$ um mapeamento onde X é o espaço de entrada e Ω representa o espaço de característica.

No novo espaço de características ainda será empregada a ideia de hiperplano separador para classificar os dados de acordo com uma função de classificação que dependa somente dos dados de entrada, como o que foi apresentado nas seções anteriores. Assim, não será mais construído um hiperplano no espaço original dos dados de entrada, mas sim alguma superfície de representação mais complexa (Ω). É por esse motivo que James et al. (2013) descrevem que a SVM é considerada uma abordagem de extrema flexibilidade com relação à quantidade de hipóteses dos dados de entrada, mas com baixa interpretabilidade quanto a forma que a superfície de separação dos dados possuirá.

Proposto em 1992, o *Truque do Kernel* foi inicialmente elaborado tendo como base o classificador de máxima margem rígida (3.7), isto é,

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w \\ \text{s.a.} \quad & y_i (w^T \phi(x_i) + b) \geq 1, \quad i = 1, 2, \dots, n. \end{aligned} \tag{3.23}$$

sendo $w \in \mathbb{R}^n$, $b \in \mathbb{R}$, $y_i \in \mathbb{R}$, $x_i \in \mathbb{R}^p$ e a função ϕ responsável por fazer o mapeamento do espaço de entrada para o espaço de características.

Por sua vez, a formulação de SVM amplamente mais utilizada, e que será abordada neste trabalho, é a que possui como base o classificador de máxima margem suave (3.17), pois a regularização possibilitará que a função de mapeamento ϕ consiga separar os dados do espaço de entrada linearmente, a saber,

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^n \xi_i \\ \text{s.a.} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, n. \end{aligned} \quad (3.24)$$

sendo $C \in \mathbb{R}$ um parâmetro, $w \in \mathbb{R}^n$, $b \in \mathbb{R}$, $y_i \in \mathbb{R}$, $x_i \in \mathbb{R}^p$, $\xi_i \in \mathbb{R}$ e a função ϕ responsável por fazer o mapeamento do espaço de entrada para o espaço de características.

Motivados pelas mesmas justificativas das seções anteriores, além do custo computacional para problemas quando n é muito grande, utiliza-se a formulação dual do problema (3.24) para determinar a função de classificação, a saber,

$$\begin{aligned} \text{minimize } f(\alpha) = \quad & \frac{1}{2} \alpha^T P \alpha - \alpha^T e \\ \text{s.a.} \quad & \sum_{i=1}^n \alpha^T y = 0 \\ & 0 \leq \alpha \leq C \end{aligned} \quad (3.25)$$

em que $\alpha \in \mathbb{R}^n$, $y \in \mathbb{R}^n$, e a matriz $P \in \mathbb{R}^{n \times n}$ definida como

$$P = \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle, \quad i = 1, 2, \dots, n. \quad (3.26)$$

e C é um parâmetro.

De acordo com Cristianini, Shawe-Taylor et al. (2000), os resultados que são obtidos no desenvolvimento de classificador de máxima margem rígida e suave, principalmente de existência de solução e unicidade, se mantêm para o problema de SVM (3.24), mesmo que aplicando a transformação ϕ nos vetores x_i . A diferença entre o problema dual margem suave (3.22) e o problema dual (3.25), é que os produtos internos entre os vetores x (no espaço original) que cada elemento da matriz P carrega envolvem agora as transformações ϕ .

Note que na matriz P da função objetivo (3.25) temos a presença de produtos internos $\langle \phi(x_i), \phi(x_j) \rangle$ no cálculo de seus elementos. Por ser um produto interno entre dois vetores, o valor resultante indicará o quão próximos estão os vetores. É devido ao

aparecimento destes produtos internos em (3.26) que elabora-se o *Truque de Kernel*, vez que pode-se trocar o referido produto interno por uma função *Kernel*, que acumulará os produtos internos no espaço de características, aplicada aos mesmos pontos x_i e x_j , evitando a construção explícita de um espaço de características dimensão n .

Para isso, defini-se a função *kernel* como a seguir:

Definição 1 Considere $\chi \in \mathbb{R}^n$. Denomina-se função *kernel* $K : \chi \times \chi \rightarrow \mathbb{R}$ caso exista um mapeamento ϕ do conjunto χ para um espaço de Hilbert H , ou seja $\phi : \chi \rightarrow H$, de forma que

$$K(a, b) = \langle \phi(a), \phi(b) \rangle,$$

para todo $a, b \in \chi$.

Desta forma, pode-se definir a função de classificação de SVM, a qual é baseada na mesma função de classificação dos métodos anteriores, dependente somente dos parâmetros $y_i \in Y$ com $Y \in \mathbb{R}^n$, $x_i \in X$ com $X \in \mathbb{R}^{n \times p}$ e da função *kernel* K , como se segue.

$$\begin{aligned} f(x) &= \text{sinal}(w^{*T} \phi(x_i) + b^*) \\ &= \text{sinal}\left(\sum_{i=1}^n y_i \alpha_i^* K(x, x_i) + b^*\right) \\ &= \begin{cases} \text{classe positiva, se } \sum_{i=1}^N y_i \alpha_i^* K(x, x_i) + b^* > 0 \\ \text{classe negativa, se } \sum_{i=1}^N y_i \alpha_i^* K(x, x_i) + b^* < 0 \end{cases} \end{aligned} \quad (3.27)$$

para qualquer $x \in \mathbb{R}^n$. de forma similar a equação (3.13), o valor do parâmetro b^* pode ser encontrado usando a função *kernel* como a seguir:

$$b^* = \frac{1}{|S|} \sum_{j \in S} \left(y_j - \sum_{i,j=1}^n \alpha_i^* y_i K(x_i, x_j) \right), \quad (3.28)$$

onde $S = \{j \in \{1, \dots, n\} : 0 < \alpha_j^* < C\}$ é o conjunto de índices, $|S|$ indica quantos elementos o conjunto S possui (SVENSÉN; BISHOP, 2007).

Algumas das funções kernel mais utilizadas na literatura para SVM, além de outros modelos de aprendizado de máquinas e reconhecimento de padrões, são descritas na Tabela 1. Cada uma delas possuem parâmetros que devem ser fornecidos pelo usuário e são também indicados na tabela.

Tabela 1 – Algumas funções *kernel*

Nome	Função(a,b)	Parâmetros
Linear	$a^T b$	-
Polynomial	$(a^T b + p)^d$	grau $d \in \mathbb{N}$, $p \in \mathbb{R}_+$
Gaussian	$\exp(-\sigma \ a - b\ ^2)$	$\sigma \in \mathbb{R}_+$
Exponential	$\exp(\sigma a^T b)$	$\sigma \in \mathbb{R}_+$
Sigmoid	$\tanh(ka^T b + \theta)$	k e $\theta < 0$

3.4 Considerações Finais do Capítulo

Neste capítulo foi apresentado o método de SVM, ao qual ilustrou-se suas ideias e destacou-se possíveis formulações. Em um problema de classificação binário, a essência deste método é definir um classificador de máxima margem entre os vetores suporte. Os vetores suporte são encarados como fundamentais no método de SVM, uma vez que são os pontos do espaço amostral mais importantes para construir o classificador. Em um aspecto geral, o método de SVM pode ser considerado bem flexível em relação a entrada de dados (quanto as hipóteses do conjunto de dados para ajuste), entretanto a interpretabilidade é reduzida a medida que o problema torna-se não linear.

A formulação enfatizada neste capítulo é a abordagem dual do problema de otimização, principalmente considerando variáveis de folga e o truque de *kernel* para os casos não lineares. Será a partir da abordagem dual do problema de SVM que o algoritmo proposto se baseará, sendo a proposta aplicada para casos em que os conjuntos de dados sejam linearmente separáveis ou não.

4 Método Proposto

Neste capítulo, será detalhado o método proposto, denominado DiagSVM, para o treinamento de Support Vector Machines (SVM). O DiagSVM é um framework que combina técnicas de aproximação diagonal da matriz Hessiana com métodos de busca por um zero de uma função para resolver problemas de SVM em larga escala. O objetivo principal do DiagSVM é reduzir o custo computacional do treinamento de SVM, mantendo a precisão e a capacidade de generalização do modelo.

O desenvolvimento do DiagSVM foi motivado pelos desafios computacionais associados ao treinamento de SVMs com conjuntos de dados de alta dimensão. A complexidade computacional dos métodos tradicionais de SVM é quase cúbica ($O(n^2m)$) em relação ao número de amostras, o que torna o treinamento impraticável para grandes volumes de dados. O DiagSVM busca mitigar este problema através de uma abordagem baseada em aproximações e decomposições, que permite resolver o problema de otimização de forma mais eficiente.

A Figura 4 apresenta um diagrama geral do framework DiagSVM. O framework é composto por duas etapas principais: a etapa de aproximação e a etapa de busca por um zero de uma função. Na etapa de aproximação, a matriz Hessiana do problema de SVM é aproximada por uma matriz diagonal utilizando técnicas esféricas e quasi-Cauchy. Na etapa de busca por um zero de uma função, os subproblemas resultantes são resolvidos utilizando métodos como bisseção, regra falsi e ponto fixo.

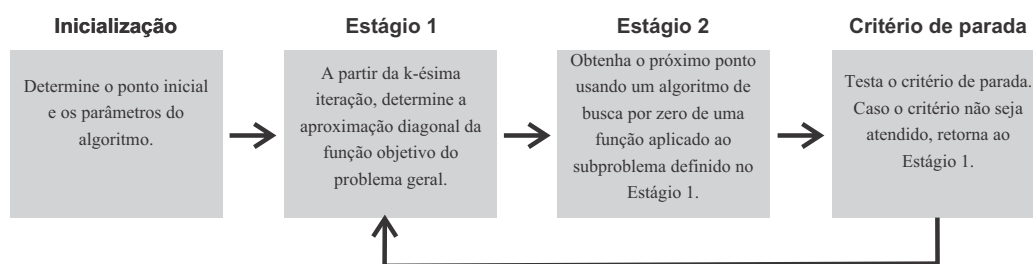


Figura 4 – Diagrama de blocos do DiagSVM

A seguir, descrevemos com mais detalhes as características do nosso esquema. Na Seção 4.1, apresentamos a estrutura de aproximação usada no primeiro estágio do framework e quais métodos usamos para a aproximação quadrática separável. Na Seção 4.2, definimos os métodos de busca por um zero de uma função que resolvem o problema estudado, apresentando alguns exemplos e referências. Na Seção 4.3, definimos a formulação final do framework e apresentamos seu algoritmo.

4.1 Primeiro Estágio do framework: Aproximações Quadráticas Separáveis

Nosso estudo para encontrar uma maneira de aproximar a matriz Hessiana da função objetivo do problema (3.25) por uma matriz diagonal teve como motivação o trabalho de [Snyman e Hay \(2001\)](#), aplicado a problemas de otimização irrestritos. Nesta seção iremos abordar os dois métodos utilizados até o momento no primeiro estágio do framework: Aproximação Quadrática Esférica e Aproximação quasi-Cauchy.

4.1.1 Aproximação Quadrática Esférica

O método de máximo declive continua sendo um dos procedimentos mais fundamentais para minimizar uma função diferenciável com várias variáveis. O método proposto por Cauchy continua sendo a base de vários procedimentos de solução baseados em gradiente ([BAZARAA; SHERALI; SHETTY, 2013](#)). O desempenho do método de gradiente descendente é, no entanto, decepcionante em comparação com outros métodos de busca linear (*line search*) e que fazem uso somente de informações de primeira ordem (gradiente) ([SNYMAN; HAY, 2001](#)).

[Snyman e Hay \(2001\)](#), afirmam que apesar de utilizar o que parece ser a "melhor" direção de busca, ou seja, aquela que dá a máxima taxa de decréscimo no ponto de aplicação, o método não é realmente eficaz na maioria dos problemas. O método de gradiente descendente geralmente funciona muito bem durante os estágios iniciais do processo de otimização, dependendo do ponto de inicialização. No entanto, quando está se aproximando de um ponto estacionário, o método geralmente se comporta mal, dando passos pequenos e quase ortogonais.

Entre os métodos que usam apenas informações de gradiente e realizam sucessivas buscas lineares, um dos métodos mais populares é provavelmente o método de gradiente conjugado apresentado em [Fletcher e Reeves \(1964\)](#). Métodos somente de gradiente, como o método de Fletcher-Reeves ([FLETCHER; REEVES, 1964](#)), continuam sendo de grande importância porque se tornam indispensáveis quando o tamanho do problema (número de variáveis) se torna muito grande. Quando o número de variáveis é grande, métodos de segunda ordem usando informações Hessianas, ou tentando construir a matriz Hessiana (como métodos quase-Newton), tornam-se impraticáveis devido ao tamanho da matriz ([BAZARAA; SHERALI; SHETTY, 2013](#)).

Para exemplificar, o método clássico de gradiente descendente é aplicado ao seguinte problema de otimização irrestrito:

$$\min \{f(x) : x \in \mathbb{R}^n\}, \quad (4.1)$$

onde $f: \mathbb{R}^n \rightarrow \mathbb{R}$ é a função objetivo e x um vetor de n componentes. Assumimos que f seja diferenciável tal que exista $\nabla f(x)$ para qualquer $x \in \mathbb{R}^n$. Assim, a solução é definida como x^* .

Com isso, os passos do algoritmo de descida mais íngreme (DI) aplicado ao problema (4.1), pode ser apresentado da seguinte forma:

Passo 1: Especifique a tolerância de convergência ϵ_g ; selecione um ponto inicial x^0 ; e defina $k = 1$. Com isso é dado início ao procedimento principal.

Passo 2: Se $\|\nabla f(x^{k-1})\| \leq \epsilon_g$, então $x^* = x^k$ e pare; caso contrário defina $d^k = -\nabla f(x^{k-1})$.

Passo 3: (Passo de busca linear) Defina θ_k tal que $f(x^{k-1} + \theta_k d^k) = \min_{\theta} f(x^{k-1} + \theta d^k)$ sujeito $\theta \geq 0$.

Passo 4: Defina $x^k = x^{k-1} + \theta_k d^k$, $k = k + 1$ e retorne ao passo 2.

Pode ser visto que, se o método de gradiente descendente for aplicado a uma função quadrática positiva definida da forma $f(x) = (1/2)x^T A x + b^T x + c$, então a sequência $\{f(x^k)\} \rightarrow f(x^*)$, o qual é o valor ótimo do problema (4.1).

Neste contexto, [Snyman e Hay \(2001\)](#) apresentaram um algoritmo baseado somente em gradiente, que em termos de armazenamento (apenas vetores precisam ser armazenados) e eficiência computacional, podem ser considerados uma alternativa aos métodos baseados em gradiente de segunda ordem. O método aplica a técnica de gradiente descendente em sucessivas aproximações esféricas quadráticas da função objetivo de um problema quadrático, de modo que nenhuma busca linear seja necessária para resolver o problema de minimização. [Snyman e Hay \(2001\)](#), mostram ainda que o método é convergente quando aplicado a funções quadráticas positivas definidas gerais.

O método de gradiente descendente quadrático esférico (DIQE) proposto em [Snyman e Hay \(2001\)](#), que enquadra-se em uma classe de métodos de otimização aproximada sequencial, gera uma sequência de subproblemas de otimização quadráticos $P[k]$, $k = 0, 1, 2, \dots$, onde cada subproblema é resolvido gerando uma sequência de soluções aproximadas x^{k+1} . Mais especificamente, em cada ponto x^k , o subproblema construído $P[k]$ é definido como

$$\min_x \bar{f}_k(x), \quad (4.2)$$

onde a função objetivo aproximada $\bar{f}_k(x)$ é dada por

$$\bar{f}_k(x) = \frac{1}{2}(x - x^k)^T Z_k(x - x^k) + \nabla f(x^k)^T(x - x^k) + f(x^k), \quad (4.3)$$

com $Z_k = \text{diag}(z_k, z_k, \dots, z_k) = z_k I$. A solução desse problema é denotado por x^{*k} , e para a construção do próximo subproblema $P[k+1]$, é usado $x^{k+1} := x^{*k}$. Para encontrar o valor do termo z_k , [Snyman e Hay \(2001\)](#) afirmam que $\bar{f}_k(x^k)$ deve interpolar $f(x)$ em ambos x^k e x^{k-1} . Essa condições implicam que para $k = 1, 2, \dots$

$$z_k = \frac{2[f(x^{k-1}) - f(x^k) - \nabla f(x^k)^T(x^{k-1} - x^k)]}{\|x^{k-1} - x^k\|^2}. \quad (4.4)$$

Então, [Snyman e Hay \(2001\)](#) afirmam que para uma grande classe de problemas, a sequência $x^0, x^1, \dots$, tenderá à solução do problema original (4.1), isto é

$$\lim_{k \rightarrow \infty} x^k = x^*. \quad (4.5)$$

Para os subproblemas $P[k]$ que sejam convexos, ou seja, $z_k > 0$, a solução ocorre quando $\nabla \bar{f}_k = 0$, isto é, onde

$$\nabla f(x^k) + z_k I(x - x^k) = 0.$$

A solução para o subproblema, x^{*k} , é portanto dado por

$$x^{*k} = x^k - \frac{\nabla f(x^k)}{z_k}. \quad (4.6)$$

Claramente, a solução para o problema quadrático esférico está ao longo de uma linha que passa por x^k na direção do gradiente descendente. O método DIQE pode ser formalmente declarado de acordo com os seguintes passos:

Passo 1: Especifique a tolerância de convergência ϵ_g , ϵ_x , e o tamanho do passo $d > 0$; selecione um ponto inicial x^0 ; defina $z_0 = \|\nabla f(x^0)\|/d$; e defina $k = 1$. Com isso é dado início ao procedimento principal.

Passo 2: Se $\|\nabla f(x^{k-1})\| \leq \epsilon_g$, então $x^* = x^{k-1}$ e para; caso contrário defina

$$x^k = x^{k-1} - \frac{\nabla f(x^{k-1})}{z_{k-1}}.$$

Passo 3: Se $\|x^k - x^{k-1}\| > d$, então defina

$$x^k = x^k - d \frac{\nabla f(x^{k-1})}{\|\nabla f(x^{k-1})\|};$$

senão se $\|x^k - x^{k-1}\| \leq \epsilon_x$, então $x^* = x^k$ e para.

Passo 4: Encontre z_k pela equação (4.4).

Passo 5: Defina $k = k + 1$ e retorne ao passo 2.

O controle do tamanho do passo é introduzido no algoritmo acima por meio da especificação de um limite de passo d e do teste no Passo 3. A escolha de z_0 garante que, para $P[0]$, a solução x^1 esteja a uma distância d de x^0 na direção do gradiente descendente. Além disso, no Passo 4, caso $z_0 < 0$, [Snyman e Hay \(2001\)](#) definem que $z_k = 10^{-60}$, assim é garantido que a função objetivo aproximada seja sempre definida positiva.

4.1.2 Aproximação Quasi-Cauchy

A fim de elucidar a flexibilidade do framework para o treinamento de SVM, procuramos na literatura outras formas de encontrar z_k . Para tal, como segunda alternativa de aproximação quadrática separável no primeiro estágio do framework, foi utilizada a relação *quasi-Cauchy* proposta em [Zhu, Nazareth e Wolkowicz \(1997\)](#) e descrita a seguir.

A relação *quasi-Cauchy* corresponde a relação *quasi-Newton* "fraca" de [Dennis Jr e Wolkowicz \(1993\)](#), onde uma restrição adicional substitui a matriz completa por uma matriz diagonal. Mais especificamente, dado o problema (4.1), a relação *quasi-Newton* "fraca" corresponde a

$$s_k^T M_k s_k = s_k^T y_k,$$

onde $s_k = x^k - x^{k-1}$, $y_k = \nabla f(x^k) - \nabla f(x^{k-1})$ e $M_k \in \mathbb{R}^{n \times n}$ é uma aproximação da Hessiana da função f em x^k . Como em [Zhu, Nazareth e Wolkowicz \(1997\)](#), uma das formas de atualizar a matriz M_k é dada como a seguir:

$$M_k = M_{k-1} + \frac{s_k^T y_k - s_k^T M_{k-1} s_k}{(s_k^T M_{k-1} s_k)^2} M_{k-1} s_k s_k^T M_{k-1}, \quad (4.7)$$

onde M_{k-1} é positiva definida. A condição $s_k^T y_k > 0$ implica que M_k é também positiva definida. Novamente, s_k e y_k são usados explicitamente e um armazenamento $O(n^2)$ é requerido.

Se M_{k-1} for escolhido para ser uma matriz diagonal positiva definida, a saber D_{k-1} , a equação (4.7) pode ser restrita a atualizar apenas os elementos diagonais de M_k , produzindo uma matriz diagonal positiva definida, D_k .

Em [Zhu, Nazareth e Wolkowicz \(1997\)](#), a relação *quasi-Cauchy* é definida como

$$s_k^T D_k s_k = s_k^T y_k,$$

onde D_k deve ser uma matriz diagonal. Com isso, somente armazenamento $O(n)$ é requerido para as atualizações diagonais. Em [\(ZHU; NAZARETH; WOLKOWICZ, 1997\)](#) as atualizações da matriz diagonal D_k acontecem de duas formas: por uma

constante múltiplo da matriz identidade ou por um vetor correspondente aos elementos de uma matriz diagonal. Iremos considerar a aproximação diagonal onde um escalar é múltiplo da matriz identidade, o qual possui relação com a conhecida matriz de escala Oren-Luenberger, ou seja,

$$D_k = (s_k^T y_k / s_k^T s_k) I, \quad (4.8)$$

sendo I a matriz identidade. Nota-se que essa é precisamente a única matriz que seria obtida da relação *quasi-Cauchy* com a restrição adicional de que a matriz diagonal é formada por uma escalar vezes a matriz identidade, ou seja, os elementos diagonais da Hessiana aproximada D_k são iguais. Assim, matrizes escalares derivadas da relação *quasi-Cauchy* são generalizações naturais da escala de Oren-Luenberger (ZHU; NAZARETH; WOLKOWICZ, 1997).

Nota-se que a aproximação *quasi-Cauchy* se assemelha a aproximação esférica apresentada através da equação (4.4), onde z_k é um escalar que compõem os autovalores de uma matriz diagonal. De fato, a equação (4.8) pode ser obtida através da equação (4.3), como é mostrado em Wilke, Kok e Groenwold (2010).

Wilke, Kok e Groenwold (2010) definem, baseado em Zhu, Nazareth e Wolkowicz (1997), o resultado obtido em (4.4) como uma aproximação diagonal baseada em informações do histórico do valor da função, onde z_k é obtido forçando $\bar{f}(x^{k-1}) = f(x^{k-1})$, e $\bar{f}$ é definida em (4.3).

Wilke, Kok e Groenwold (2010) definem também o que consideram aproximações diagonais baseadas em informações do gradiente, onde a derivada de (4.3) em relação a x é dada por

$$\nabla \bar{f}_k(x) = \nabla f(x^k) + Z_k(x - x^k). \quad (4.9)$$

Em $x = x^k$, os gradientes da função f e da função aproximada $\bar{f}_k$ coincidem e, neste caso, a Hessiana aproximada Z_k de $\bar{f}_k$ é escolhida para combinar informações adicionais. Wilke, Kok e Groenwold (2010) consideram, novamente, apenas o caso de uma aproximação quadrática esférica onde $Z_k = z_k I$. Então z_k é obtido combinando o vetor gradiente em x^{k-1} . Como apenas um único parâmetro z_k está disponível, as n componentes dos respectivos vetores gradientes são combinados utilizando a ideia de mínimos quadrados.

O erro de mínimos quadrados é dado por:

$$E = (\nabla \bar{f}(x^{k-1}) - \nabla f(x^{k-1}))^T (\nabla \bar{f}(x^{k-1}) - \nabla f(x^{k-1})), \quad (4.10)$$

no qual, após substituir 4.9 em 4.10, obtém-se

$$E = (\nabla f(x^k) + z_k(x^{k-1} - x^k) - \nabla f(x^{k-1}))^T (\nabla f(x^k) + z_k(x^{k-1} - x^k) - \nabla f(x^{k-1})). \quad (4.11)$$

A minimização do erro de mínimos quadrados E com relação a z_k resulta em

$$\begin{aligned} \frac{\partial E}{\partial z_k} &= (\nabla f(x^k) + z_k(x^{k-1} - x^k) - \nabla f(x^{k-1}))^T (x^{k-1} - x^k) \\ &\quad + (x^{k-1} - x^k)^T (\nabla f(x^k) + z_k(x^{k-1} - x^k) - \nabla f(x^{k-1})) = 0, \end{aligned} \quad (4.12)$$

assim,

$$z_k = \frac{(x^k - x^{k-1})^T (\nabla f(x^k) - \nabla f(x^{k-1}))}{(x^k - x^{k-1})^T (x^k - x^{k-1})} \quad (4.13)$$

A equação (4.13) corresponde a equação (4.8), obtida por meio da aproximação esférica.

4.1.3 Adaptando a Aproximação Quadrática Separável ao Framework

Baseado no algoritmo DIQE, a proposta do framework é aproximar a função objetivo do problema de otimização gerado pelo método do SVM (3.25) por uma função objetivo separável. Sendo assim, usando as ideias introduzidas em (SNYMAN; HAY, 2001), convertamos a equação (4.3) para o formato que estamos procurando resolver de acordo com a função objetivo da equação (3.25), obtendo o seguinte resultado:

$$\begin{aligned} \bar{f}_k(x) &= \frac{1}{2}(x - x^k)^T Z_k(x - x^k) + \nabla f(x^k)^T (x - x^k) + f(x^k) \\ &= \nabla f(x^k)^T x - \nabla f(x^k)^T x^k + \frac{1}{2} \left[(x - x^k)^T Z_k x - (x - x^k)^T Z_k x^k \right] \\ &= \nabla f(x^k)^T x - \nabla f(x^k)^T x^k + \frac{1}{2} \left[x^T Z_k x - x^{kT} Z_k x - x^T Z_k x^k + x^{kT} Z_k x^k \right] \\ &= \nabla f(x^k)^T x - \nabla f(x^k)^T x^k + \frac{1}{2} x^T Z_k x - \frac{1}{2} x^{kT} Z_k x - \frac{1}{2} x^T Z_k x^k + \frac{1}{2} x^{kT} Z_k x^k. \end{aligned} \quad (4.14)$$

Como estamos interessados em minimizar a função $\bar{f}_k(x)$, os termos $\nabla f(x^k)^T x^k$ e $\frac{1}{2} x^{kT} Z_k x^k$, os quais são constantes, podem ser suprimidos, e de (4.14), obtemos que

$$\bar{f}_k(x) = \frac{1}{2} x^T Z_k x - \left[-\nabla f(x^k) + Z_k x^k \right]^T x. \quad (4.15)$$

A equação (4.15) juntamente com as restrições do problema (3.25), correspondem aos subproblemas quadráticos separáveis gerados em cada iteração do framework. As funções em (4.3) e (4.15) serão sempre separáveis e convexas enquanto os termos z_k forem estritamente positivos, tendo assim uma matriz Hessiana positiva definida e diagonal.

Com isso, finalizamos a introdução das duas abordagens utilizadas no primeiro estágio do framework, a saber, Aproximação Diagonal Esférica e Aproximação Quasi-Cauchy, que são sumarizadas a seguir:

- Em [Snyman e Hay \(2001\)](#), a Aproximação Diagonal Esférica é definida seguindo a equação (4.3). Eles afirmam que forçando $\bar{f}_k(x^{k-1}) = f(x^{k-1})$, é possível obter z_k como a seguir:

$$z_k = \frac{2[f(x^{k-1}) - f(x^k) - \nabla f(x^k)^T(x^{k-1} - x^k)]}{\|x^{k-1} - x^k\|^2}.$$

Assim, a aproximação separável da função objetivo em (4.3) e (4.15), usam um escalar como múltiplo da matriz identidade.

- Em [Zhu, Nazareth e Wolkowicz \(1997\)](#), a Aproximação Quasi-Cauchy pode ser obtida através da seguinte equação: $D_k = (s_k^T y_k / s_k^T s_k) I$, onde $s_k = x^k - x^{k-1}$, $y_k = \nabla f(x^k) - \nabla f(x^{k-1})$ e I é a matriz identidade. D_k é a conhecida matriz diagonal de escala de Oren-Luenberger. Reescrevendo a equação (4.8) no mesmo formato da equação (4.4), podemos obter z_k da Aproximação Quasi-Cauchy da seguinte forma:

$$z_k = \frac{(x^k - x^{k-1})^T (\nabla f(x^k) - \nabla f(x^{k-1}))}{(x^k - x^{k-1})^T (x^k - x^{k-1})}$$

Na próxima subseção, introduzimos algumas direções de pesquisa e outras em andamento para serem aplicadas ainda no primeiro estágio do framework.

4.1.4 Outras Opções para o Primeiro Estágio do Framework

Existem ainda outras alternativas de aproximação quadrática separável para serem aplicadas no primeiro estágio do framework, que incluem estratégias de atualização da matriz diagonal e técnicas de região de confiança para melhorar os resultados já obtidos.

4.1.4.1 Aproximação Quasi-Newton

Em [Andrei \(2019\)](#), o autor propõe uma atualização diagonal quasi-Newton. Os elementos da matriz diagonal que aproximam a Hessiana são determinados minimizando tanto o tamanho da mudança da estimativa anterior quanto o traço da atualização, sujeito à equação da secante fraca. Em seus experimentos, o autor [Andrei \(2019\)](#) mostra que a atualização diagonal quasi-Newton é mais eficiente e robusta que métodos baseados em gradiente (método de gradiente descendente). Já em [Andrei \(2020\)](#), é sugerido um novo método quasi-Newton onde as atualizações da matriz Hessiana são diagonais e os elementos diagonais são determinados por diferenças finitas diretas ou centrais.

Para resolver o problema irrestrito (4.1), o método quasi-Newton introduzido em Davidon (1959), recebe um ponto inicial $x_0 \in \mathbb{R}^n$ e uma aproximação inicial da Hessiana simétrica e definida positiva $B_0 \in \mathbb{R}^{n \times n}$, gerando então uma sequência

$$x_{k+1} = x_k + \gamma_k d_k, \quad (4.16)$$

com $k = 0, 1, \dots$, e $d_k \in \mathbb{R}^n$ a direção de busca calculada através da seguinte sistema linear:

$$B_k d_k = -g_k, \quad (4.17)$$

sendo g_k é a derivada da função f em x_k . Em (4.17), $B_k \in \mathbb{R}^{n \times n}$ é uma aproximação quasi-Newton da Hessiana $\nabla^2 f(x_k)$. Para garantir que a aproximação quasi-Newton contenha as informações corretas da curvatura, B_{k+1} , a atualização de B_k deve satisfazer a equação secante ou quasi-Newton (ANDREI, 2019):

$$B_{k+1} s_k = y_k. \quad (4.18)$$

Geralmente, para considerações práticas, o tamanho do passo γ_k em (4.16) é determinado utilizando método de Wolfe (ANDREI, 2019).

Assim como em Zhu, Nazareth e Wolkowicz (1997), em Andrei (2019) a proposta utiliza a relação *quasi-Newton* "fraca" de Dennis Jr e Wolkowicz (1993) com matrizes restritas a serem diagonais e minimizando o traço da matriz que aproxima a Hessiana. Assim é considerado

$$B_k = \text{diag}(b_k^1, \dots, b_k^n), \quad (4.19)$$

onde $b_k^i > 0$, para todo $i, \dots, n$, como a aproximação diagonal da matriz Hessiana da função f na iteração k .

Assim, o algoritmo proposto em Andrei (2019) resolve o problema (4.1) utilizando a mesma estratégia utilizada em Zhu, Nazareth e Wolkowicz (1997). Com isso em mente, os estudos para expandir as possibilidades de aproximações separáveis utilizadas no primeiro estágio do framework continuam nesta direção.

4.1.4.2 Técnicas de Região de Confiança e Conservadorismo

Estratégias são necessárias para garantir que os métodos de aproximação quadrática separáveis convergirão (SNYMAN; WILKE et al., 2005). Uma prioridade é portanto garantir que as aproximações construídas produzem uma solução suficientemente precisa. Com esse objetivo, Goldfeld, Quandt e Trotter (1966) propuseram restringir o

tamanho do passo com base na validade da aproximação sobre um domínio, que mais tarde foi batizado em [Sorensen \(1982\)](#) como os conhecidos métodos de região de confiança. Como uma medida alternativa para os métodos de região de confiança, conservadorismo foi adotado em [Svanberg \(2002\)](#).

O conservadorismo requer que a cada iteração, a atualização proposta seja viável, assim como, possua uma melhora com relação a iteração anterior. O benefício de ambas as abordagens é que fortes características de convergência para abordagens de aproximação sequencial podem ser provadas, embora para classes de funções restritas ([SNYMAN; WILKE et al., 2005](#)).

Em um esforço para manter o conservadorismo dentro do contexto de abordagens que fazem uso de gradiente, [Wilke, Kok e Groenwold \(2010\)](#) sugeriram que a derivada direcional do problema real na solução aproximada proposta ao longo da direção do passo de atualização deveria ser negativa. Em cada iteração k , a solução encontrada x^{*k} é obtida através do passo de atualização $x^{*k} - x^k$ a partir da solução anterior x^k . Esta atualização representa a descida de $f(x)$ ao longo da direção $x^{*k} - x^k$ se

$$\nabla^T f(x^{*k})(x^{*k} - x^k) \leq \nabla^T \bar{f}(x^{*k})(x^{*k} - x^k). \quad (4.20)$$

Consequentemente, qualquer aproximação baseada somente em gradientes pode ser definida como conservadora se (4.20) for válida ([WILKE; KOK; GROENWOLD, 2010](#)).

Essa definição de conservadorismo baseado em métodos somente em gradientes é semelhante em intenção, ao conservadorismo no qual se baseia no valor da função ([WILKE; KOK; GROENWOLD, 2010](#)), onde o mesmo requer que o valor da função $f(x^{*k})$ seja melhor que o da iteração anterior. Em métodos baseados somente em gradientes, utilizam apenas atualizações x^{*k} para as quais a medida de qualidade do problema original,

$$\nabla^T f(x^k)(x^k - x^k),$$

sejam menores ou igual a medida de qualidade do problema aproximado

$$\nabla^T \bar{f}(x^{*k})(x^{*k} - x^k),$$

serão aceitos. Embora nenhuma prova formal seja apresentada neste trabalho, [Wilke, Kok e Groenwold \(2010\)](#) mostraram que essa definição de conservadorismo garante convergência para certas classes de funções, por exemplo, funções convexas suaves.

4.2 Segundo Estágio do Framework: Métodos de Busca por um Zero de uma Função

Uma vez que no primeiro estágio foi gerado um subproblema quadrático separável para o SVM, apresentamos nessa seção um novo problema de otimização, conhecido na literatura como Problema da Mochila Quadrática Separável (PMQS). No nosso trabalho ele é um caso específico em que a matriz Hessiana é diagonal e o Problema da Mochila se torna separável. Por isso, apresentamos uma metodologia para resolver esse problema que é de formato semelhante ao SVM diagonalizado no primeiro estágio.

Os métodos de busca por um zero de uma função utilizados neste trabalho são aqueles cujo objetivo é resolver o problema PMQS, que por sua vez tem a seguinte forma:

$$(PMQS) : \begin{cases} \text{minimize } f(x) = \frac{1}{2}x^T Px - a^T x \\ \text{sujeito à} & b^T x = c \\ & l \leq x \leq u, \end{cases} \quad (4.21)$$

onde $x \in \mathbb{R}^n$ é a variável de decisão; $P = \text{diag}(p)$ é uma matriz diagonal positiva definida; a, b são vetores em $\mathbb{R}^n$, c é um escalar; e $l < u$ são os limites inferiores e superiores. No problema (4.21), reformulando x como uma função dependente de λ , temos um problema de encontrar o multiplicador de Lagrange da restrição de igualdade. A prova dos três resultados teóricos a seguir pode ser encontrada em Münnich, Sachs e Wagner (2012).

Teorema 1 Um vetor $x^* \in \mathbb{R}^n$ é um mínimo do problema (4.21), se e somente se, existe um multiplicador de Lagrange $\lambda_* \in \mathbb{R}$, $v^* \in \mathbb{R}_+^n$ e $w^* \in \mathbb{R}_+^n$ tal que

$$\nabla L_x(x^*, \lambda_*, v^*, w^*) = \nabla f(x^*) + \lambda_* \nabla g(x^*) + (\nabla r(x^*))' v^* + (\nabla s(x^*))' w^* = 0 \quad (4.22)$$

e além disso,

$$v_i r_i(x_i^*) = 0, \quad i = 1, \dots, n, \quad (4.23)$$

$$w_i s_i(x_i^*) = 0, \quad i = 1, \dots, n, \quad (4.24)$$

onde $f(x) = \frac{1}{2}x^T Px - a^T x$, $g(x) = b^T x - c$, $r(x) = l - x$ e $s(x) = x - u$.

Lema 4.2.1 De acordo com a condições do Teorema de KKT, (4.22) é equivalente a

$$\left. \begin{aligned} 0 &\geq \frac{p_i u_i - a_i}{b_i} + \lambda_*, & \text{se } x_i^* = u_i \\ 0 &= \frac{p_i x_i^* - a_i}{b_i} + \lambda_*, & \text{se } x_i^* \in (l_i, u_i) \\ 0 &\leq \frac{p_i l_i - a_i}{b_i} + \lambda_*, & \text{se } x_i^* = l_i, \end{aligned} \right\} \quad (4.25)$$

para todo $i = 1, 2, \dots, n$.

Motivado por (4.25), a função x é definida como dependente da variável λ .

$$x : \mathbb{R} \rightarrow \mathbb{R}^n, \lambda \mapsto x(\lambda)$$

com

$$x_i(\lambda) = \begin{cases} l_i, & \text{se } \lambda \geq \frac{a_i - p_i l_i}{b_i} \\ \frac{a_i - \lambda b_i}{p_i}, & \text{se } \frac{a_i - p_i u_i}{b_i} < \lambda < \frac{a_i - p_i l_i}{b_i} \\ u_i, & \text{se } \lambda \leq \frac{a_i - p_i u_i}{b_i}. \end{cases} \quad (4.26)$$

Teorema 2 Um vetor $x^* \in \mathbb{R}^n$ é a única solução do problema de otimização (4.21), se e somente se, existe $\lambda_* \in \mathbb{R}$ tal que $x(\lambda_*)$, definido em (4.26), satisfazendo

$$g(x(\lambda_*)) = 0. \quad (4.27)$$

A partir do desenvolvimento acima, observe que para resolver o PMQS, podemos usar métodos de busca que determinam zeros da função unidimensional não diferenciável $g: \mathbb{R} \rightarrow \mathbb{R}$.

Segundo [Bretthauer e Shetty \(2002\)](#), duas abordagens básicas para resolver o problema (4.21) são os métodos de busca pelo multiplicador e relaxação ou o método de fixação de variável. Esses métodos podem ser vistos como dois diferentes membros da família de algoritmos que operam mantendo um subconjunto das condições de Karush-Kuhn-Tucker (KKT) satisfeitas, e eles demandam um esforço computacional a cada iteração para manter as condições KKT satisfeitas. O método de busca pelo multiplicador mantém todas as condições KKT durante suas iterações, exceto a restrição simples $b^T x = c$, e isso corresponde a uma restrição complementar de folga. Uma solução ótima é obtida quando essas duas condições são satisfeitas. No método de fixação de

variável, todas as condições KKT são mantidas durante as iterações, exceto o limite nas variáveis. A solução ótima é obtida uma vez que os limites das variáveis são satisfeitos (BRETTHAUER; SHETTY, 2002).

Bretthauer e Shetty (2002) apresentaram um algoritmo que resolve o problema (4.21) em que a restrição de igualdade é uma inequação no formato $b^T x \leq c$, utilizando uma busca unidimensional pelo multiplicador ótimo de Lagrange da restrição simples.

Baseado nos estudos realizados sobre o problema PMQS, para que a proposta pudesse resolver bem o problema de treinamento do SVM, ainda foram desenvolvidos durante esse projeto de pesquisa em um trabalho conjunto, outros métodos para serem aplicados no segundo estágio do framework, a saber, método do ponto fixo (ALVES et al., 2023) e suas extensões assim como métodos baseados na aproximação da função max, como o método *softplus* (DUGAS et al., 2000).

4.2.1 Método do Ponto Fixo

Em Alves et al. (2023), é apresentada uma estratégia baseada no algoritmo do ponto fixo de Münnich, Sachs e Wagner (2012) para resolver o problema (4.21). De acordo com a equação (4.26), os autores definem os seguintes conjuntos:

$$\begin{aligned} I_\ell(\lambda) &:= \left\{ i \in \{1, 2, \dots, n\} : \lambda \geq \frac{a_i - p_i l_i}{b_i} \right\} \\ I_u(\lambda) &:= \left\{ i \in \{1, 2, \dots, n\} : \lambda \leq \frac{a_i - p_i u_i}{b_i} \right\} \\ I_{eq}(\lambda) &:= \{1, 2, \dots, n\} \setminus (I_\ell(\lambda) \cup I_u(\lambda)). \end{aligned} \quad (4.28)$$

Então, temos que

$$\begin{aligned} g(x(\lambda)) &= b^T x(\lambda) - c \\ &= \sum_{i \in I_\ell(\lambda)} b_i l_i + \sum_{i \in I_u(\lambda)} b_i u_i + \sum_{i \in I_{eq}(\lambda)} b_i \frac{a_i - \lambda b_i}{p_i} - c \\ &= \sum_{i \in I_\ell(\lambda)} b_i l_i + \sum_{i \in I_u(\lambda)} b_i u_i + \sum_{i \in I_{eq}(\lambda)} \left(\frac{b_i a_i}{p_i} - \frac{\lambda b_i^2}{p_i} \right) - c \\ &= \sum_{i \in I_\ell(\lambda)} b_i l_i + \sum_{i \in I_u(\lambda)} b_i u_i + \sum_{i \in I_{eq}(\lambda)} \frac{b_i a_i}{p_i} - \lambda \sum_{i \in I_{eq}(\lambda)} \frac{b_i^2}{p_i} - c. \end{aligned} \quad (4.29)$$

Definindo $g(x(\lambda)) = 0$, conseguimos encontrar o valor de λ de forma iterativa de acordo com a expressão abaixo:

$$\lambda \sum_{i \in I_{eq}(\lambda)} \frac{b_i^2}{p_i} = \sum_{i \in I_\ell(\lambda)} b_i l_i + \sum_{i \in I_u(\lambda)} b_i u_i + \sum_{i \in I_{eq}(\lambda)} \frac{b_i a_i}{p_i} - c,$$

isto é,

$$\lambda = \frac{\sum_{i \in I_\ell(\lambda)} b_i l_i + \sum_{i \in I_u(\lambda)} b_i u_i + \sum_{i \in I_{eq}(\lambda)} \frac{b_i a_i}{p_i} - c}{\sum_{i \in I_{eq}(\lambda)} \frac{b_i^2}{p_i}}. \quad (4.30)$$

Assim, é definido a seguinte equação

$$F: \mathbb{R} \rightarrow \mathbb{R},$$

$$\lambda \mapsto F(\lambda) = \frac{\sum_{i \in I_\ell(\lambda)} b_i l_i + \sum_{i \in I_u(\lambda)} b_i u_i + \sum_{i \in I_{eq}(\lambda)} \frac{b_i a_i}{p_i} - c}{\sum_{i \in I_{eq}(\lambda)} \frac{b_i^2}{p_i}}. \quad (4.31)$$

Com isso, $g(x(\lambda)) = 0$ se, e somente se, $\lambda = F(\lambda)$.

A fórmula (4.31) aparece em [Kiwiel \(2008\)](#) com um passo intermediário no algoritmo de fixação de variável. Além disso, seguindo [Kiwiel \(2008\)](#), assumimos que para todo $k \in \mathbb{N}$, se $I_{eq}(\lambda_k) \neq \emptyset$, então $I_{eq}(\lambda_{k+1}) \neq \emptyset$.

Abaixo descrevemos formalmente os passos do algoritmo de ponto fixo (abreviado como FP). Por uma questão de simplicidade, para $k \in \mathbb{N}$, denotamos $I_\ell(\lambda^k) = I_\ell^k$, $I_u(\lambda^k) = I_u^k$ e $I_{eq}(\lambda^k) = I_{eq}^k$.

Passo 1: Defina $\lambda_0 \in \mathbb{R}$.

Passo 2: Obtenha $I_\ell^k, I_u^k, I_{eq}^k$ de acordo com (4.28).

Passo 3: Encontre λ_{k+1} com

$$\lambda_{k+1} = \frac{\sum_{i \in I_\ell^k} b_i l_i + \sum_{i \in I_u^k} b_i u_i + \sum_{i \in I_{eq}^k} \frac{b_i a_i}{p_i} - c}{\sum_{i \in I_{eq}^k} \frac{b_i^2}{p_i}} \quad (4.32)$$

Passo 4: Se $\lambda_{k+1} = \lambda_k$, então pare. Caso contrário, defina $k = k + 1$ e retorne ao Passo 2.

4.2.2 Método do Ponto Fixo Acelerado

Como mencionado em [Walker e Ni \(2011\)](#), métodos de aceleração podem aliviar a convergência lenta de métodos baseados em iteração do ponto fixo aplicados a algumas classes de problemas. Nosso interesse aqui está em dois métodos de aceleração específicos. O primeiro originou-se do trabalho de [Anderson \(1965\)](#), a que nos referimos como aceleração de Anderson, e a segunda resultou do trabalho de [Aitken \(1925\)](#), a que nos referimos como aceleração de Aitken. Nas subseções seguintes, descrevemos ambos os métodos que podem ser incorporados no algoritmo FP.

4.2.2.1 Aceleração de Anderson

A aceleração de Anderson define um vetor de pesos $\alpha \in \mathbb{R}^m$. Esses pesos são determinados por meio do seguinte problema de otimização:

$$\min_{\alpha} \|G_k \alpha\|_2 \quad s.t. \sum_{i=0}^m \alpha_i = 1, \quad (4.33)$$

onde G é encontrado da forma a seguir. Vamos considerar $\lambda = F(\lambda)$ de acordo com a Equação (4.31) e $g = F(\lambda) - \lambda$. Então, $G_k = [g_{k-m}, \dots, g_i]$, onde $g_k = F(\lambda_k) - \lambda_k$.

Com esses pesos podemos criar a expressão para a próxima iteração como:

$$\lambda_{k+1} = \sum_{i=0}^m \alpha_i F(\lambda_{k-m+i}) \quad (4.34)$$

Provas de convergência e o algoritmo da aceleração de Anderson podem ser vistos em Walker e Ni (2011). Já o algoritmo acelerado incorporado ao algoritmo FP pode ser visto em Alves et al. (2023).

4.2.2.2 Aceleração de Aitken

Como em Baumann e Klymak (2021), a ideia da aceleração de Aitken é alterar o fator de relaxação (e, assim, o tamanho do passo de iteração) baseado nas informações das iterações passadas.

Seguindo Baumann e Klymak (2021), vamos considerar a sequência de escalares $\{\lambda_k\}_{k=0}^{\infty}$ que converge linearmente para seu ponto fixo $\hat{\lambda}$. Isso implica que para um k grande:

$$\frac{\hat{\lambda} - \lambda_{k+1}}{\hat{\lambda} - \lambda_k} \approx \frac{\hat{\lambda} - \lambda_{k+2}}{\hat{\lambda} - \lambda_{k+1}}. \quad (4.35)$$

Abaixo, reorganizamos a equação (4.35) para fornecer uma fórmula que prevê o ponto fixo usado como a iteração subsequente.

$$\lambda_{k+1} = \lambda_k - \frac{(\lambda_{k+1} - \lambda_k)^2}{\lambda_{k+2} - 2\lambda_{k+1} + \lambda_k}. \quad (4.36)$$

O algoritmo acelerado com aceleração de Aitken incorporado ao algoritmo FP pode ser visto em Alves et al. (2023).

4.2.3 Métodos baseados em aproximação da função $\max$

Primeiramente, com base em (KIWIEL, 2008), a solução primal lagrangiana do problema (4.21) pode ser escrita como

$$x(\lambda) = \min \left\{ \max \left[l, \frac{(a - \lambda b)}{p} \right], u \right\}, \quad (4.37)$$

onde o mínimo e o máximo são obtidos por componentes, e assim como mostrado no Teorema 2, as condições de KKT para (4.21) podem ser reescritas como

$$g(\lambda) := b^T x(\lambda) - c. \quad (4.38)$$

Considere a função f_{max} denifida como

$$f_{max} = \max\{0, a\} \quad (4.39)$$

onde $a \geq 0$. Essa função também é conhecida como função retificadora (*rectifier function*) e é bastante utilizada em problemas de aprendizado de máquina.

Com isso em mente, nas próximas subseções descrevemos algumas abordagens que aproximam a função max no formato da equação (4.39), por uma função separável, possibilitando um novo método para resolver o problema PMQS. Tal abordagem também é resultado de um trabalho conjunto (ALVES; SILVA; SANTOS, 2023) e também será introduzida nas próximas seções.

4.2.3.1 Aproximação *softplus*

A função *softplus* foi inicialmente proposta em Dugas et al. (2000) e até agora é usada principalmente como uma aproximação continuamente diferenciável do problema (4.39) em redes neurais profundas (ZHENG et al., 2015). A função mapeia os elementos de $\mathbb{R}$ para seu subconjunto positivo $\mathbb{R}_+$. Como em (WIEMANN; KNEIB; HAMBUCKERS, 2021), definimos a seguinte versão generalizada da função *softplus*:

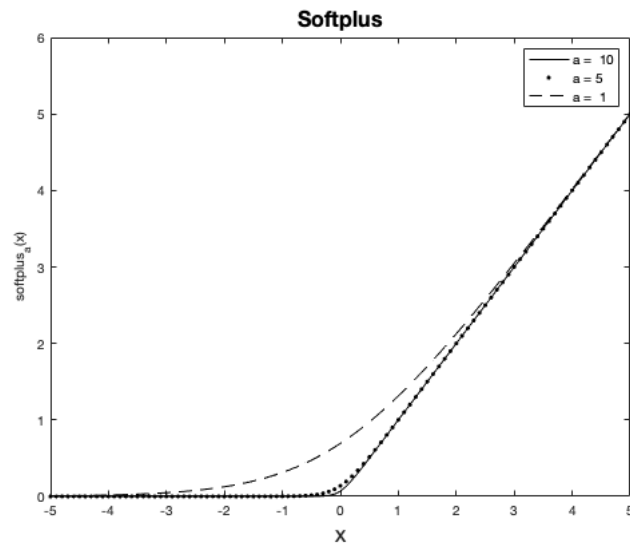
$$softplus_a(x) = \frac{\log(1 + \exp(ax))}{a} \quad (4.40)$$

onde a variável $a > 0$ é um parâmetro adicional. Definindo $a = 1$ a função é reduzida à sua forma simples. A figura 5 mostra a função *softplus* sujeita a diferentes valores do parâmetro a . Seguindo Wiemann, Kneib e Hambuckers (2021), concluímos que introduzindo o parâmetro a , o erro de aproximação pode ser controlado em relação à função *softplus*, pois como ilustrado a seguir, para todo $\epsilon > 0$, existe algum $\alpha > 0$ tal que

$$0 \leq softplus_a(x) - \max(0, x) \leq \log(2)/a < \epsilon,$$

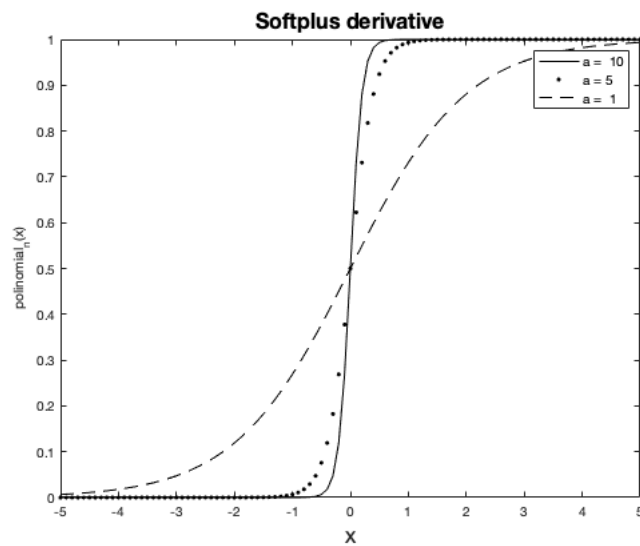
vale para todo $x \in \mathbb{R}$. O maior erro de aproximação está em $x = 0$ conforme indicado visualmente pela Figura 5. Além disso, pode-se observar na Figura 5, que a função *softplus* segue a função identidade muito de perto no domínio positivo e se aproxima rapidamente de zero no domínio negativo para $x \rightarrow -\infty$. Ao aumentar o parâmetro a , tal comportamento pode ser ainda mais acentuado.

A derivada da equação (4.40) pode ser definida como

Figura 5 – Função *Softplus*

$$\sigma_a(x, a) = \frac{\exp(xa)}{\exp(xa) + 1} \quad (4.41)$$

A figura 6 mostra a derivada da função *softplus* sujeita a diferentes valores do parâmetro a .

Figura 6 – Derivada da função *softplus*

Em [Wiemann, Kneib e Hambuckers \(2021\)](#), os autores também investigam a estabilidade numérica da função *softplus*.

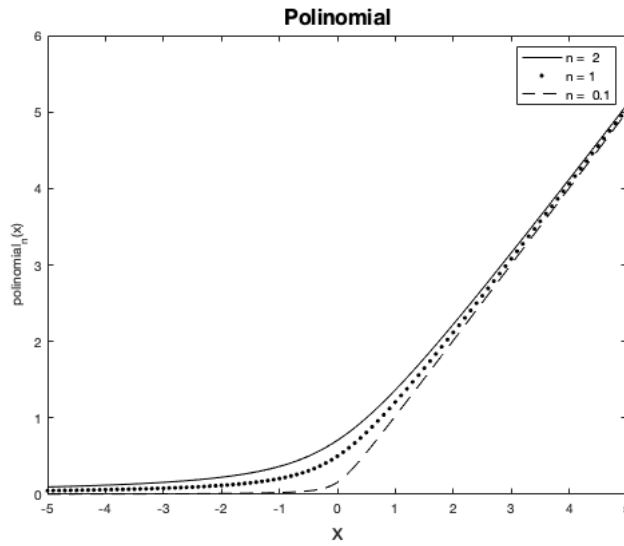


Figura 7 – Função polinomial

4.2.3.2 Aproximação polinomial

Assim como a função *softplus*, nesta seção apresentamos uma função polinomial na forma da equação (4.42), que corresponde a uma aproximação da função *max* (4.39). A Figura 7 exibe o gráfico da função polinomial para diferentes valores do parâmetro n e $\beta = 2$.

$$polynomial_{\beta}(x) = \frac{1}{2}(x + (x^{\beta} + n)^{1/\beta}) \quad (4.42)$$

Para o problema PMQS que queremos resolver, fixamos o valor do parâmetro $\beta = 2$ para todos os casos, já que esse foi o valor que apresentou melhores resultados nos testes numéricos.

A derivada da equação (4.42) pode ser definida como

$$\sigma_b(x, n) = \frac{1}{2}(1 + (x^{\beta} + n)^{-1/\beta}x) \quad (4.43)$$

.

A Figura 8, mostra a derivada da função polinomial sujeita a diferentes valores do parâmetro n e $\beta = 2$.

4.2.3.3 Aplicando aproximação da função max no problema PMQS

Para aplicar as funções *softplus* e polinomial apresentadas nas subseções anteriores ao problema (4.21), é necessário uma reformulação do problema (4.37). Para simplificar a operação, renomeamos $x(\lambda)$ para $proj(x)$ e definimos $x = \frac{(a-\lambda b)}{p}$ no problema (4.37). Finalizamos colocando o problema resultante em função da função *max* (4.39),

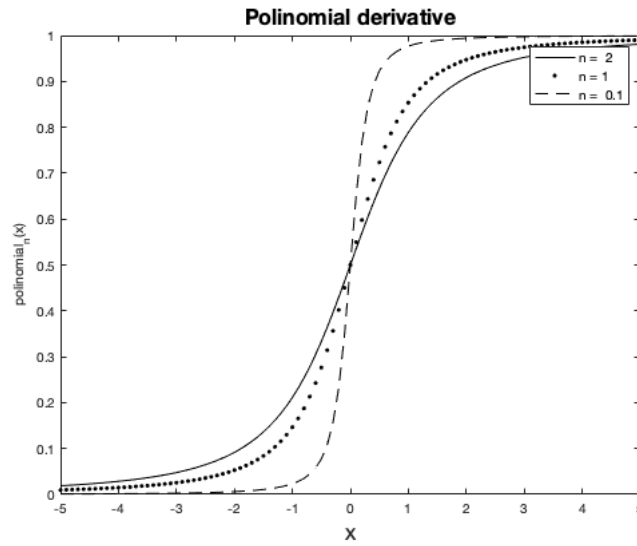


Figura 8 – Derivada da função polinomial

ficando:

$$\begin{aligned}
 proj(x) &= \min\{\max[l, x], u\} \\
 &= \max\{l, \min[x, u]\} \\
 &= \max\{l, -\max[-x, -u]\} \\
 &= \max\{l, x - \max[0, -u + x]\} \\
 &= l + \max\{0, x - \max[0, -u + x] - l\} \\
 &= l + \max\{0, x - l - \max[0, -u + x]\}
 \end{aligned} \tag{4.44}$$

Definimos as aproximações da função $\max$ como φ , neste caso podemos substituir a função $\max$ na equação (4.44) por φ resultando em

$$proj(x) = l + \varphi\{0, x - l - \varphi[0, -u + x]\}. \tag{4.45}$$

Agora, podemos definir a seguinte expressão:

$$\tilde{g}(\lambda) := b^T proj(x) = c \tag{4.46}$$

Como a equação (4.46) será diferenciável, podemos usar sua derivada como uma aproximação derivada da equação (4.38) para definir o seguinte algoritmo baseado no método de Newton.

Passo 1: Defina $\lambda_0 \in \mathbb{R}$ e $k = 0$.

Passo 2: Obtenha $\tilde{g}(\lambda_k)$, $\Delta\tilde{g}(\lambda_k)$.

Passo 3: $\lambda_{k+1} = \lambda_k - \frac{\tilde{g}(\lambda_k)}{\Delta \tilde{g}(\lambda_k)}$ Se $\lambda_{k+1} - \lambda_k < \epsilon$, então pare. Caso contrário, defina $k = k + 1$ e retorne ao Passo 2.

Mais detalhes teóricos a respeito dos métodos baseado em aproximações da função *max* aplicados a resolução do problema PMQS, podem ser vistos em (ALVES; SILVA; SANTOS, 2023).

4.2.3.4 Extensões para aproximação da função max

Em Sivapalan et al. (2022), os autores propuseram uma rede neural *lightweight* baseada em uma implementação de ponto fixo da rede para reduzir a complexidade do modelo. Eles usam uma função de ativação *Sigmoid* para ajustar os pesos dos neurônios de acordo com a equação (4.41). Como a equação (4.41) não é fácil de implementar em um ambiente de ponto fixo, os autores a traduziram para uma versão aproximada e rápida ($\tilde{\sigma}$): a função de ativação Elliott (ELLIOTT, 1993).

$$\tilde{\sigma}(x, e) = \frac{0.5x}{e + |x|} + 0.5 \quad (4.47)$$

A Figura 9, exibe a função Elliot sujeita a diferentes valores do parâmetro e .

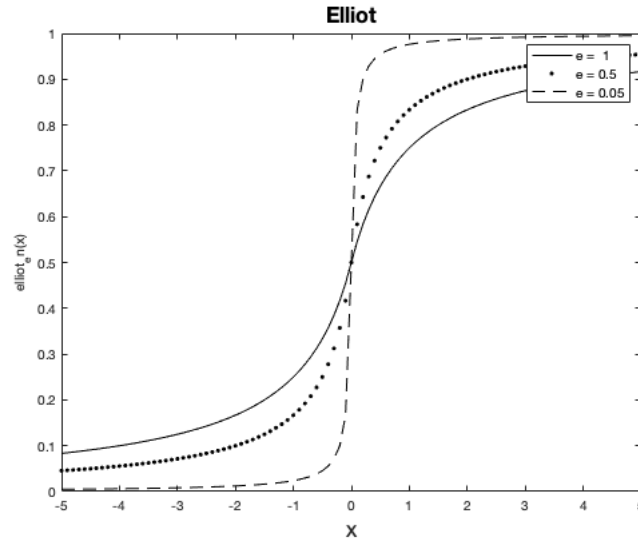


Figura 9 – Função Elliot

Experimentos preliminares realizados em Alves, Silva e Santos (2023), mostraram que a função Elliot proporcionou resultados com um tempo computacional melhor que utilizando a função (4.41), sem comprometer os resultados obtidos.

4.2.4 Revisão dos métodos utilizados no segundo estágio do framework

Sumarizamos abaixo alguns trabalhos que apresentam estratégias de busca por um zero de uma função e foram utilizados durante os experimentos numéricos presentes

neste trabalho. Outros também são apresentados como possibilidades para serem implementados no segundo estágio do framework proposto:

1. Em Münnich, Sachs e Wagner (2012), uma estratégia de ponto fixo para um problema de amostragem estratificada (não quadrática mas separável) é comparada com métodos clássicos da literatura: Bissecção, Regula falsi e Secante.
2. Em Cominetti, Mascarenhas e Silva (2014), um método do tipo Newton para PMQS é estudado e comparações com os métodos Secante, *Median Search* e fixação de variáveis são incluídos.
3. Em Kiwiel (2008), um algoritmo básico de fixação de variável é desenvolvido para resolver o PMQS e propõem várias opções de implementação para a solução do problema reduzido e suas atualizações.
4. Em Alves et al. (2023), desenvolvemos um método de ponto fixo para resolver PMQS com base em (MÜNNICH; SACHS; WAGNER, 2012).
5. Ainda seguindo Alves et al. (2023), desenvolvemos uma extensão para o método de ponto fixo aplicando técnicas de aceleração.
6. Em Alves, Silva e Santos (2023), desenvolvemos um algoritmo baseado na aproximação da função *max* aplicado ao problema PMQS reformulado.

4.3 Estrutura do Algoritmo Proposto

Usando os resultados das seções anteriores, estamos prontos para formular o algoritmo do framework para resolver o problema SVM QP apresentado na Seção 3.3 de fundamentação teórica:

1. Obtenha os dados rotulados para treinar o modelo: $X \in \mathbb{R}^{n \times m}$ e $y \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$.
2. Defina os parâmetros $C > 0$ e função Kernel K .
3. Defina α^0 , z^0 e $k = 1$.
4. Na primeira fase do framework, dado o problema (3.25), aplicamos uma das duas aproximações separáveis apresentadas na Seção 4.1: Aproximação Quadrática Esférica (equação 4.4) ou Aproximação Quasi-Cauchy (equação 4.13).

- Como resultado da aproximação separável, no primeiro estágio do framework, obteremos o seguinte problema:

$$\begin{aligned} &\text{minimize } f(\alpha) = \frac{1}{2}\alpha^T Z_k \alpha - \left[-\nabla f(\alpha^k) + Z_k \alpha^k \right]^T \alpha \\ &\text{subject to } y^T \alpha = 0 \\ &0 \leq \alpha \leq Ce, \end{aligned} \quad (4.48)$$

5. No segundo estágio do framework, obtenha o próximo ponto α^k usando um algoritmo de busca por um zero de uma função para resolver o problema (4.48)
6. Se $\alpha^k = \alpha^{k-1}$ então $\alpha^* = \alpha^{k-1}$ e pare; caso contrário, defina $k = k + 1$ e retorne para o primeiro estágio do algoritmo no passo 4.

4.4 Implementação e Detalhes Técnicos

Abaixo, detalhamos cada passo da implementação do algoritmo apresentado na seção anterior:

Passo 1: Obtenha os dados rotulados para treinar o modelo: $X \in \mathbb{R}^{n \times m}$ e $y \in \mathbb{R}^n, y_i \in \{-1, 1\}$.

Passo 2: Defina os parâmetros $C > 0$ e a função Kernel K .

Passo 3: Gere o problema SVM QP:

$$\begin{aligned} &\text{minimize } f(\alpha) = \frac{1}{2}\alpha^T P \alpha - \alpha^T e \\ &\text{sujeito a } y^T \alpha = 0 \\ &0 \leq \alpha \leq Ce, \end{aligned} \quad (4.49)$$

onde, $P \in \mathbb{R}^{n \times n}$ de acordo com (3.26), $\alpha \in \mathbb{R}^n$, $e \in \mathbb{R}^n$ é um vetor de uns, y é o mesmo da Passo 1.

Passo 4: Defina $k = 0$ e $\alpha^k = e$;

Defina $\nabla f^k = P\alpha^k - e$, $z^k = \|\nabla f^k\|e$, and $a^k = -\nabla f^k + z^k * \alpha^k$;

Passo 5: Use um algoritmo de determinação de zero de um função para resolver o problema (4.50).

$$\begin{aligned} &\text{minimize } \tilde{f}(\alpha^k) = \frac{1}{2}\alpha^T Z^k \alpha^k - \alpha^k a^k \\ &\text{sujeito a } y^T \alpha^k = 0 \\ &0 \leq \alpha^k \leq Ce \end{aligned} \quad (4.50)$$

onde $Z^k = \text{diag}(z^k)$.

Passo 6: Defina $k = k + 1$ e $\alpha^k = \tilde{f}(\alpha^{k-1})$, sendo $\tilde{f}$ a função objetivo do problema (4.50);

Do Teorema 2, α^k é uma solução única do problema (4.50), se λ^k satisfaz (4.27). Assim, usamos λ^k como critério de parada, de modo que, se $\lambda^k - \lambda^{k-1} < \epsilon$ então $\alpha^* = \alpha^k$ e PARE; caso contrário, continue no próximo passo.

Passo 7: Se estiver usando SQA:

Defina $f^k = \frac{1}{2}\alpha^{kT}P\alpha^k - \alpha^{kT}e$ e $\nabla f^k = P\alpha^k - e$;

Defina $s = \|x^k - x^{k-1}\|^2$ e $v = 2(f^{k-1} - f^k - \nabla f^k s)$;

Defina $z^k = \max(10^{-20}, \frac{s}{v})$, $a^k = -\nabla f^k + z^k * \alpha^k$, e $z^k = z^k e$.

Vá para o Passo 5.

Se estiver usando QCA:

Defina $\nabla f^k = P\alpha^k - e$;

Defina $s = x^k - x^{k-1}$ e $v = \nabla f^k - \nabla f^{k-1}$;

Defina $z^k = \max(10^{-20}, \frac{s^T v}{s^T s})$, $a^k = -\nabla f^k + z^k * \alpha^k$, e $z^k = z^k e$.

Vá para o Passo 5.

4.4.1 Complexidade do Algoritmo Proposto

Considere o algoritmo da Seção 4.4. Os Passos 1 e 2 definem algumas variáveis necessárias para executar o algoritmo, que podem ser processadas em tempo constante ($O(1)$). No Passo 3, o problema SVM QP (4.50) é gerado. Como usamos a função Kernel RBF (ver Tabela 1), a complexidade de tempo desta etapa pode atingir $O(n^2m)$ (CRISTIANINI; SHAW-TAYLOR et al., 2000), onde n é o número de pontos de dados, e m é a dimensão do espaço de entrada. No entanto, na literatura, há algumas pesquisas de implementação rápida para a função de kernel RBF (YOKOTA; BARBA; KNEPLEY, 2010; QIU; LANE, 2005). No Passo 4, o processamento de maior custo computacional é a operação matriz-vetor, que requer $O(n^2)$ de complexidade. No Passo 5, usamos métodos de busca por um zero de uma função $O(n)$ para resolver cada subproblema. O Passo 6 verifica os critérios de parada, que requerem apenas tempo constante. O Passo 7 gera a nova aproximação da função objetivo geral a partir do resultado da última iteração. A aproximação diagonal esférica e a aproximação quase-Cauchy requerem $O(n^2)$ de complexidade pois, de forma semelhante ao Passo 4, o processamento mais custoso é uma operação matriz-vetor.

4.5 Considerações Finais do Capítulo

Neste capítulo, discutiu-se a proposta de framework para resolver o problema de treinamento de SVM. O framework é baseado no método de aproximação quadrática

separável, onde duas alternativas podem ser aplicadas no problema original de treinamento do SVM no primeiro estágio do framework: Aproximação quadrática esférica e aproximação quasi-Cauchy.

No segundo estágio do framework, são usados métodos de busca por um zero de uma função para resolver os subproblemas gerados. Além dos métodos tradicionais do estado da arte, também propomos algumas abordagens que combinadas com as aproximações quadráticas separáveis, tornam-se boas alternativas quando aplicados ao treinamento de SVM.

5 Experimentos numéricos

Para ilustrar o comportamento numérico do algoritmo proposto, DiagSVM, foi realizado duas metodologias de experimentos numéricos. Na primeira, utilizamos algumas bases de dados de benchmark para problemas de classificação binária. Na segunda, formulamos um problema de classificação e diagnósticos de doenças cutâneas por imagem.

A fim de testar a performance do DiagSVM, comparações foram feitas com o algoritmo Libsvm (CHANG; LIN, 2011). Usamos a pesquisa em grade (*grid search*) para encontrar os parâmetros ideais do SVM. Para compor a *grid search*, selecionamos quatro valores diferentes de C (1E+0, 1E+1, 1E+2, 1E+3) e σ (1E-1, 1E-2, 1E-3, 1E-4). Assim, para cada problema de classificação foram realizadas 16 (4 x 4) combinações.

A Tabela 2 exibe a descrição das métricas de desempenho utilizadas neste estudo. Na tabela 2, VP , VN , FP e FN referem-se a verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo, respectivamente.

Tabela 2 – Métricas de performance

Nome da Medida	Descrição	Fórmula
Acurácia (A_c)	É definido como o número de amostras classificadas corretamente em relação ao número total de amostras testadas.	$A_c = \frac{VP+VN}{VP+VN+FP+FN}$
Especificidade (S_p)	É a razão entre o número de decisões VN do classificador e os N casos reais testados.	$S_p = \frac{VN}{VN+FP}$
Sensibilidade (S_e)	É a razão entre o número de decisões VP do classificador e os P casos reais testados.	$S_e = \frac{VP}{VP+FN}$

Os experimentos numéricos são organizados em seções da seguinte forma: Na Seção 5.1, descrevemos os conjuntos de dados selecionados para os experimentos de benchmark assim como apresentamos seus resultados obtidos. Já na seção 5.1.2, formulamos o problema de Classificação de Lesões Cutâneas (CLC), descrevemos as bases de dados utilizadas e apresentamos os resultados obtidos.

5.1 Problemas de benchmark

Para os problemas de benchmark, o framework foi testado em um laptop Intel Core i5 1.6 GHz, 8 GB de RAM, processador dual-core, com código implementado em MATLAB 2021b.

É importante destacar que usaremos os resultados obtidos nos problemas de benchmark para selecionar a melhor combinação do DiagSVM e aplicá-la ao problema CLC (Seção 5.2 a seguir).

5.1.1 Descrição do conjunto de dados de benchmark

Um dos conjuntos de dados que usamos é o MNIST de [LeCun, Cortes e Burges \(2010\)](#). Esse banco de dados para reconhecimento de dígitos manuscritos tem sido amplamente utilizado para validar novas técnicas de aprendizado de máquina, como em [Cominetti, Mascarenhas e Silva \(2014\)](#). [Baldominos, Saez e Isasi \(2019\)](#) apresentaram uma pesquisa que resume as principais contribuições de ponta relatadas no MNIST.

Todos os outros conjuntos de dados foram extraídos do site oficial do Libsvm (<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>). Os autores do Libsvm já fornecem os dados em formato apropriado para classificação, e também oferecem uma função para transformar os dados em matrizes e vetores com dimensões relativas às amostras n e atributos p .

Os conjuntos de dados foram escalados linearmente para os intervalos $[-1, 1]$ ou $[0, 1]$. A principal vantagem do dimensionamento é evitar que as faixas de valores numéricos de alguns atributos sejam muito maiores do que as faixas de valores de outros atributos. Outra vantagem é evitar dificuldades numéricas durante o cálculo ao usar a função Kernel ([HSU et al., 2003](#)). Para esses experimentos, apenas dados em escala foram usados. Nesse caso, a transformação de um atributo r de qualquer conjunto de dados pode ser descrita no intervalo $[a, b]$ na forma

$$r_{scaled} = (a - b) \frac{r - \min(r)}{\max(r) - \min(r)} + a. \quad (5.1)$$

Tabela 3 descreve os dados selecionados, ilustrando o número de amostras que o conjunto de dados possui (n) e o número de atributos que são considerados (p). A técnica de Cross-Validation (CV) será aplicada nos experimentos realizados para obtenção de um conjunto de teste disjunto. A técnica consiste em separar uma porcentagem do conjunto de dados original - neste trabalho foi utilizado 70% - para um conjunto de treinamento do modelo, enquanto o complemento é destinado à fase de teste.

A seguir, descreve-se brevemente sobre cada conjunto de dados. Em alguns

Tabela 3 – Conjunto de dados

Nome	n	p
MNIST	1568	784
heart	270	13
german numer	100	24
australian	690	14
fourclass	862	2
ionosphere	351	34
sonar	208	60
splice	1000	60

conjuntos, existem valores faltantes devido a falta de registro no estágio inicial de extração dos dados.

- **australian:** Em alguns conjuntos, existem valores faltantes devido a falta de registro no estágio inicial de extração dos dados.
- **fourclass:** conjunto de dados sintéticos que originalmente foi projetado com quatro classes, porém foi pré-processado pelos autores do LIBSVM para transformá-lo em duas classes. Sua disposição no espaço se propagam de forma irregular, com regiões desconectadas e linearmente não separáveis. Sem valores faltantes;
- **german numer:** conjunto de dados relativos a informações de dados de crédito na Alemanha. Os seus atributos são relativos a informações de pessoas, contendo informações físicas, renda, histórico de cartão, nacionalidade, entre outros, mas que são informações mais qualitativas do que quantitativas. Para o treinamento do classificador, estes dados são pré-processados pelos autores do LIBSVM para números inteiros. Sem valores faltantes;
- **heart:** um conjunto de dados que, de acordo com informações de hábitos e informações de exames, objetiva-se prever se o paciente possui ou não doença no coração. É um subconjunto de dados do conjunto original, contendo apenas treze de 75 atributos. Não possui dados faltantes;
- **ionosphere:** um conjunto de dados que é referente a captação de ondas de radares instalados no Canadá. O objetivo é identificar se possuem elétrons livres na camada ionosfera. Os sinais recebidos foram processados por uma função de autocorrelação, no qual seus argumentos eram o tempo de um pulso e o número do pulso. No sistema utilizado, foram utilizados 17 pulsos ao qual cada pulso gera um valor real e um complexo. Estes valores são resultantes da função sinal complexa eletromagnética, o que totaliza 34 atributos no conjunto. Não possui valores faltantes;

- sonar: um conjunto de dados que possui leituras de sinal de sonar, ao qual pretende-se distinguir se o sinal é referente a um material cilíndrico de metal ou rochas aparentemente cilíndricas. Não possui dados faltantes;
- splice: o objetivo deste conjunto de dados é reconhecer os dois tipos de junções splice em sequências de DNA. Esta junção é importante durante a criação da proteína, pois é um local do DNA no qual uma parte considerada supérflua é removida. Não possui dados faltantes;

O conjunto de dados MNIST é composto por 1568 imagens de número 0 a 9 (ver Figuras 10 e 11). Para utilizá-la em nossos experimentos, foi necessário um pré-processamento de cada imagem. Primeiramente, para ter um problema de classificação binária, o conjunto de dados foi dividido em duas categorias:

- Número zero (0)



Figura 10 – Número 0

- Outros números (1-9)



Figura 11 – Números entre 1 e 9

De acordo com [LeCun, Cortes e Burges \(2010\)](#), as imagens originais em preto e branco (dois níveis) do MNIST tiveram tamanho normalizado para caber em uma caixa de 20x20 pixels, preservando sua proporção. As imagens resultantes contêm níveis de cinza devido à técnica anti-aliasing usada pelo algoritmo de normalização. As imagens foram centralizadas em uma imagem de 28x28 calculando o centro de massa dos pixels e transladando a imagem para posicionar este ponto no centro do campo de 28x28. Após redimensionar a matriz 28x28, obteve-se um vetor de tamanho 784.

5.1.2 Resultados numéricos para os problemas de benchmark

Como parte dos experimentos, para o primeiro estágio do *framework*, foram utilizadas a aproximação esférica ([SNYMAN; HAY, 2001](#)) e a aproximação quasi-Cauchy ([ZHU; NAZARETH; WOLKOWICZ, 1997](#)). Para o segundo estágio do *framework*, foram

utilizados o método do Ponto Fixo (ALVES et al., 2023) e os métodos da Secante, Bissecção e Regula falsi conforme em Münnich, Sachs e Wagner (2012). Tais combinações foram desenvolvidas para exemplificar a habilidade do framework em permitir a combinação de técnicas de aproximação diagonal no primeiro estágio e qualquer outro algoritmo da literatura capaz de resolver o Problema (4.21) no segundo estágio. Ainda baseado em Snyman e Hay (2001), para o primeiro subproblema definimos $z_0 = |\nabla \tilde{f}_0|$ e α_0 qualquer ponto factível.

À medida que o CV foi aplicado, 10 sementes foram selecionadas aleatoriamente nos dados para gerar diferentes subconjuntos de dados de treinamento e teste. Assim, os resultados finais serão os valores médios dos 10 problemas gerados.

Nos experimentos, nomeamos os algoritmos para que seja mais fácil entender qual combinação foi usada no experimento. A Tabela 4 lista o que significam as siglas que compõem cada algoritmo.

Tabela 4 – Significado das siglas dos algoritmos

Sigla	Significado
Sp	Aproximação esférica
Qc	Aproximação Quasi-Cauchy
fp	Método do ponto fixo
sc	Método secante
rf	Método Regula falsi
bs	Método Bissecção
LIBSVM	<i>Library for SVM</i>

Nas Tabelas 5, 6, 7, 8 e 9, apresentamos os resultados da aproximação quasi-Cauchy com ponto fixo e método secante, aproximação quasi-Cauchy com regula falsi e bissecção, aproximação esférica com ponto fixo e método secante, aproximação esférica com regula falsi e bissecção e o método Libsvm respectivamente. Os resultados indicam que a proposta obteve resultados semelhantes ao Libsvm para todos os conjuntos de dados de benchmark. Nas tabelas, adicionamos as colunas C e σ , indicando qual valor do parâmetro o framework (DiagSVM) obteve os melhores resultados.

A tabela 5 apresenta a comparação entre a aproximação quasi-Cauchy com ponto fixo (Qc-fp) e a aproximação quasi-Cauchy com método secante (Qc-sc). Os resultados para a métrica acurácia são semelhantes para ambos os métodos em todos os conjuntos de dados. Para os conjuntos de dados mnist, heart e german, o Qc-fp apresentou melhor sensibilidade.

A tabela 6 mostra que a aproximação quasi-Cauchy com regula falsi (Qc-rf) e a aproximação quasi-Cauchy com bissecção (Qc-bs) apresentaram desempenho muito semelhantes para todas as métricas em todos os conjuntos de dados, exceto para a

Tabela 5 – Resultados da aproximação quasi-Cauchy com ponto fixo e método secante

Nome	Qc-fp					Qc-sc				
	C	σ	A_c	S_p	S_e	C	σ	A_c	S_p	S_e
mnist	1E+2	1E-2	98.74	97.73	100.00	1E+1	1E-2	98.77	99.48	98.78
heart	1E+1	1E-2	84.32	82.35	89.36	1E+0	1E-2	84.57	91.43	76.09
australian	1E+1	1E-1	85.17	79.59	91.74	1E+0	1E-2	85.65	76.85	92.93
german	1E+1	1E-2	73.73	55.65	85.41	1E+0	1E-2	74.83	56.52	78.35
fourclass	1E+2	1E+1	99.73	98.84	100.00	1E+2	1E+1	99.77	98.84	100.00
ionosphere	1E+1	1E-1	95.43	93.55	95.35	1E+1	1E-1	95.43	93.55	95.35
sonar	1E+1	1E-1	90.16	96.30	82.86	1E+1	1E-1	90.16	96.30	82.86
splice	1E+3	1E-2	80.93	92.45	80.00	1E+3	1E-2	80.85	92.45	80.00

Tabela 6 – Resultados da aproximação quasi-Cauchy com regula falsi e método de bissecção

Nome	Qc-rf					Qc-bs				
	C	σ	A_c	S_p	S_e	C	σ	A_c	S_p	S_e
mnist	1E+1	1E-2	98.77	99.48	98.78	1E+1	1E-2	98.77	99.48	98.78
heart	1E+0	1E-2	84.57	91.43	76.09	1E+0	1E-2	84.57	91.43	76.09
australian	1E+0	1E-2	85.65	76.85	92.93	1E+0	1E-2	85.65	76.85	92.93
german	1E+0	1E-2	74.83	56.52	78.35	1E+0	1E-2	75.00	56.52	78.35
fourclass	1E+2	1E+1	99.77	98.84	100.00	1E+2	1E+1	99.77	98.95	100.00
ionosphere	1E+1	1E-1	95.24	93.55	95.35	1E+1	1E-1	95.43	93.55	95.35
sonar	1E+1	1E-1	90.16	96.30	82.86	1E+1	1E-1	89.68	96.30	82.86
splice	1E+3	1E-2	80.85	92.45	80.00	1E+3	1E-2	80.68	92.16	77.61

Tabela 7 – Resultados da aproximação esférica com ponto fixo e método secante

Nome	Sp-pf					Sp-sc				
	C	σ	A_c	S_p	S_e	C	σ	A_c	S_p	S_e
mnist	1E+3	1E-2	98.94	98.98	98.70	1E+3	1E-2	98.94	98.98	98.70
heart	1E+1	1E-2	84.81	90.48	84.62	1E+1	1E-2	84.81	90.48	84.62
australian	1E+0	1E-1	85.89	79.61	92.31	1E+1	1E-1	86.38	88.37	90.08
german	1E+0	1E-2	76.20	66.67	78.05	1E+0	1E-2	76.10	67.92	78.14
fourclass	1E+0	1E+1	99.88	100.00	100.00	1E+0	1E+1	99.88	100.00	100.00
ionosphere	1E+1	1E-1	94.10	95.77	97.06	1E+1	1E-1	94.00	95.77	97.06
sonar	1E+2	1E-1	90.00	96.67	84.38	1E+3	1E-1	88.55	86.67	87.50
splice	1E+1	1E-2	79.92	84.62	77.27	1E+1	1E-2	80.00	84.62	77.27

sensibilidade na base de dados splice, onde Qc-rf obteve um resultado melhor que Qc-bs. Como esperado, os parâmetros C e σ de Qc-pf, Qc-sc, Qc-rf e Qc-bs apresentados nas Tabelas 5 e 6 foram idênticos, pois a aproximação diagonal é a mesma para todos os algoritmos.

Tabela 8 – Resultados da aproximação esférica com regula falsi e método de bissecção

Nome	Sp-rf					Sp-bs				
	C	σ	A_c	S_p	S_e	C	σ	A_c	S_p	S_e
mnist	1E+3	1E-2	98.94	98.98	98.70	1E+3	1E-2	98.94	98.98	98.70
heart	1E+1	1E-2	84.81	90.48	84.62	1E+1	1E-2	84.81	90.48	84.62
australian	1E+1	1E-1	86.33	89.53	90.91	1E+1	1E-1	86.52	89.53	90.91
german	1E+0	1E-2	76.13	67.92	78.14	1E+0	1E-2	76.13	67.92	78.14
fourclass	1E+0	1E+1	99.92	100.00	100.00	1E+0	1E+1	99.92	100.00	100.00
ionosphere	1E+1	1E-1	94.10	95.77	97.06	1E+1	1E-1	94.19	95.77	97.06
sonar	1E+2	1E-1	90.00	96.67	84.38	1E+2	1E-1	89.84	96.67	84.38
splice	1E+1	1E-2	80.00	84.62	77.27	1E+1	1E-2	80.00	84.62	77.27

Tabela 9 – Resultados do Libsvm

Nome	C	σ	A_c	S_p	S_e
mnist	1E+3	1E+1	99.34	99.23	96.25
heart	1E+0	1E-2	83.58	90.91	72.92
australian	1E+1	1E+1	86.62	83.93	88.42
german	1E+0	1E+1	76.63	72.50	75.77
fourclass	1E+3	1E-2	99.73	100.00	100.00
ionosphere	1E+1	1E-2	95.33	92.96	97.06
sonar	1E+3	1E+1	86.29	88.00	83.78
splice	1E+1	1E-1	81.86	80.95	83.64

Da mesma forma que a aproximação quasi-Cauchy, ao compararmos a aproximação esférica combinada com os algoritmos de localização de raízes nas tabelas 7 e 8, o desempenho da acurácia foi muito semelhante em todas as combinações, exceto para o conjunto de dados do sonar, onde a aproximação esférica com ponto fixo (Sp-fp), regula falsi (Sp-rf) e bissecção (Sp-bs) apresentou melhor acurácia e especificidade do que a aproximação esférica com método secante (Sp-sc). Os parâmetros C e σ foram idênticos em todas as combinações de aproximação esférica.

Comparado ao Libsvm, a diferença mais significativa nos resultados aparece na base de dados sonar, onde Qc-fp, Qc-sc, Qc-rf, Sp-pf e Sp-rf obtiveram cerca de 90% de acurácia contra 86,29% do Libsvm. O Libsvm foi superior ao solucionador SVM proposto no conjunto de dados splice e mnist com uma pequena margem.

Finalmente, na Figura 12, apresentamos um gráfico de barras com a soma da acurácia de todos os algoritmos para todos os conjuntos de dados. A figura enfatiza o que dissemos antes sobre o solucionador SVM proposto ter um desempenho semelhante ao Libsvm em relação à métrica de acurácia. Com base na Figura 12, como não há diferença significativa entre os algoritmos, escolhemos o algoritmo Qc-fp para compor o DiagSVM

nos experimentos numéricos para o problema CLC apresentado na próxima seção.

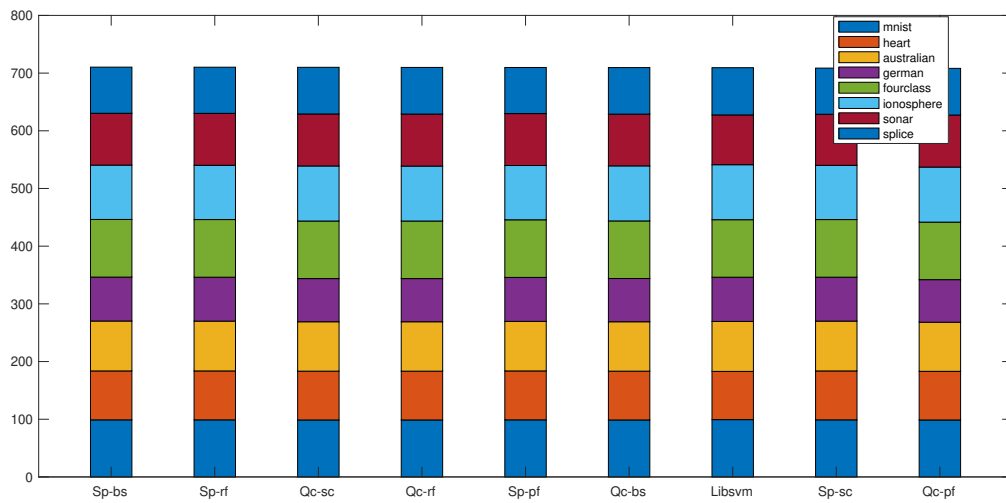


Figura 12 – Gráfico de barra com a soma das acurácias obtidas em todos os algoritmos e em todos os conjuntos de dados de benchmark.

5.2 Problema de Classificação de Lesões Cutâneas (CLC)

Para os experimentos realizados nesta seção, os algoritmos foram implementados em Python 3.8, instalado em um computador Intel Core i5-9400 CPU 2,90GHz, 24 GB de RAM; Radeon RX Vega (VEGA10, DRM 3.42.0, 5.15.0-94-genérico, LLVM 12.0.0), com 8 GB de memória.

A abordagem proposta será avaliada e validada utilizando as bases de dados descritas na Seção 5.2.1.1: conjunto de dados PH2 e ISIC2017. Cada conjunto de dados consiste em três classes de lesões. Para ambos os conjuntos de dados, manipulamos as classes para gerar dois problemas de classificação binária conforme mostrado nas tabelas 10, 11 e 12. O desempenho dos modelos de classificação é validado através de três métricas: acurácia, especificidade e sensibilidade. Todas as métricas foram descritas na Tabela 2.

5.2.1 Formulação do Problema CLC

Neste estudo, nosso foco é validar o DiagSVM apresentado nas seções anteriores, mas também contribuir com uma metodologia moderna para o CLC. A figura 13 ilustra um diagrama de blocos desta abordagem, que emprega as seguintes etapas: conjunto de dados, aumento de dados, extração de características, engenharia de recursos e classificação. Cada um deles é descrito nas subseções a seguir.

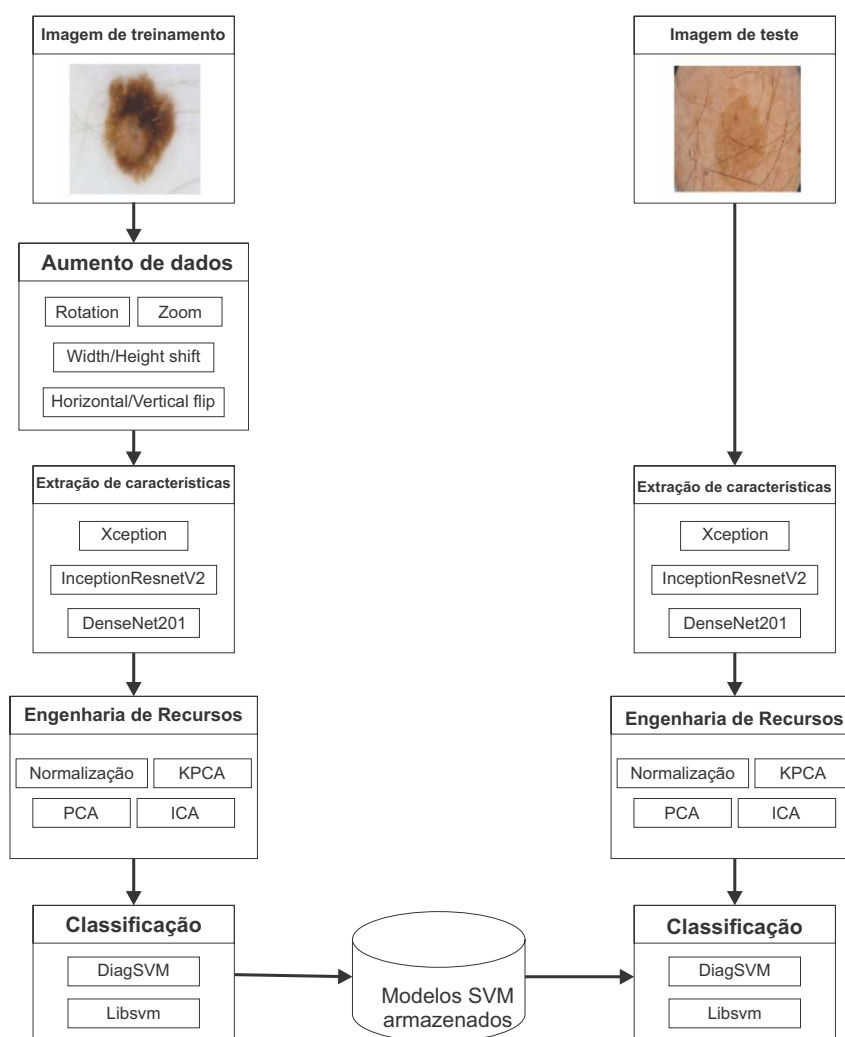


Figura 13 – Diagrama de blocos para classificação de lesões cutâneas

5.2.1.1 Conjunto de dados de imagens dermatoscópicas

Os conjuntos de dados da International Skin Imaging Collaboration (ISIC) tornaram-se um repositório de referência aos pesquisadores em aprendizado de máquina para análise de imagens médicas, especialmente na detecção de câncer de pele e avaliação de malignidades (CASSIDY et al., 2022). O conjunto de dados ISIC2017 (CODELLA et al., 2018) contém 2.000 imagens de treinamento e 600 imagens de teste, um total de 2.600 imagens. Informações básicas e metadados do paciente são fornecidos para conjuntos de treinamento e teste, indicando se a lesão pertence a um dos três grupos de classes: melanoma (ME), nevos benignos (BN) e ceratose seborreica (SK). A idade e o sexo aproximados do paciente também são fornecidos como metadados adicionais. Dentre as imagens de treinamento, 374 pertencem à classe ME, 254 à classe SK e 1372 à classe BN no formato JPG. Dentre as imagens de teste, 117 pertencem à classe ME, 90 à classe SK e 393 à classe BN, também em formato JPG. Definimos dois problemas de classificação binária para testar nossa metodologia usando o conjunto de dados ISIC2017 de acordo com a Tabela 10.

Tabela 10 – Descrição das classes usadas em problemas de classificação binária para o conjunto de dados ISIC2017 antes da etapa de aumento (augmentation).

	ME, BN - SK		ME, SK - BN	
	Classe 1 ME, BN	Classe 2 SK	Classe 1 ME, SK	Classe 2 BN
Amostras de treinamento	1746	254	628	1372
Amostras de teste	510	90	207	393

Na Tabela 10, são apresentados dois problemas de classificação binária; cada Problema compreende as classes preenchidas nas colunas Classe 1 e Classe 2. Por exemplo, o problema "ME, BN - SK" é composto pela fusão das classes ME e BN para representar a classe positiva (ver Capítulo 3), e a classe SK foi usada para representar a classe negativa. A mesma ideia se aplica ao problema "ME, SK - BN". As linhas Amostras de treinamento e Amostras de teste corresponde a quantidade de amostras que cada classe possui nas etapas de treinamento e teste.

PH2 é um banco de dados de informações sobre lesões cutâneas dermatoscópicas. O conjunto de dados PH2 compreende muitas imagens manuais de segmentação de lesões cutâneas para diagnóstico clínico e pesquisa. Dermatologistas, ou especialistas em doenças de pele, identificaram diferentes estruturas dermatoscópicas de lesões cutâneas (MENDONÇA et al., 2013). As imagens dermatoscópicas neste conjunto de dados foram obtidas nas mesmas condições através do sistema Tuebinger Mole Analyzer com ampliação de 20x. São imagens coloridas RGB de 8 bits com resolução de 768x560 pixels. O banco de dados contém 200 imagens de dermatoscopia, onde 80 são nevos comuns (NC), 80 são nevos atípicos (AN) e 40 são melanomas (ME). Como o conjunto de dados PH2 tem apenas 200 amostras, dividimos cada classe em 70% de amostras de treinamento e 30% de amostras de teste. Da mesma forma que os problemas gerados pelo conjunto de dados ISIC2017, abaixo na Tabela 11, apresentamos os dois problemas de classificação binária usando o conjunto de dados PH2 após a divisão dos dados.

Tabela 11 – Descrição das classes usadas nos problemas de classificação binária para o conjunto de dados PH2 antes da etapa de aumento (augmentation).

	CN, AN - ME		AN, ME - CN	
	Classe 1 CN, AN	Classe 2 ME	Classe 1 AN, ME	Classe 2 CN
Amostras de treinamento	112	28	84	56
Amostras de teste	48	24	36	12

Na Tabela 11, seguimos a mesma metodologia utilizada para construir os problemas da Tabela 10; dois problemas de classificação binária foram formados a partir do conjunto de dados PH2.

Como podemos ver na Tabela 10, o número de amostras na etapa de treinamento para o conjunto de dados ISIC2017 é grande, mas muito desequilibrado para todos os problemas. Em contraste, para o conjunto de dados PH2, na Tabela 11, todos os problemas possuem poucas amostras em cada classe. Na próxima etapa, apresentaremos o processo de aumento de dados (augmentation) para obter mais amostras em algumas classes de problemas afim de tornar o problema mais balanceado.

5.2.1.2 Aumento dos dados (augmentation)

O aumento de dados é comumente usado para superar o número limitado de imagens e reduzir o overfitting (SHETTY et al., 2022). Nossa metodologia utilizou inversão horizontal/vertical, mudança de largura/altura, zoom e rotação para treinar os modelos de aprendizado de máquina. A figura 14 representa as imagens de amostra após o processo de aumento utilizando inversão horizontal/vertical (SHETTY et al., 2022).

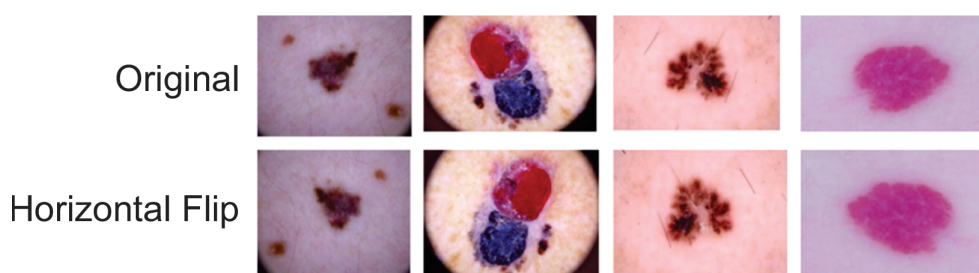


Figura 14 – Aumento de dados utilizando Horizontal Flip

Como resultado, os modelos com aumento de dados têm maior probabilidade de aprender mais recursos característicos das imagens do que os modelos sem aumento de dados. Na Tabela 12, apresentamos o número de amostras que obtivemos após o aumento dos dados para todos os problemas que realizamos. O aumento foi aplicado apenas para as amostras de treinamento.

5.2.1.3 Extração de características

CNN é um tipo específico de rede neural feed-forward com uma pilha de camadas convolucionais, cada uma seguida por camadas de agrupamento para extrair características dos dados de entrada e produzir um conjunto de mapas de características de alto nível em cada nível de convolução. As informações do mapa de características são resumidas usando camadas de pooling para reduzir os parâmetros, seguidas por uma camada totalmente conectada para produzir a classificação final (KASSANI; KASSANI,

Tabela 12 – Descrição das classes usadas em problemas de classificação binária para o conjunto de dados ISIC2017 e PH2 após a etapa de aumento.

Conjunto de dados	Classe 1	Classe 2	Número de amostras para Classe 1	Número de amostras para Classe 2
ME, BN - SK	ME, BN	SK	2046	1774
ME, SK - BN	ME, SK	BN	1985	2023
CN, AN - ME	CN, AN	ME	1556	1148
AN, ME - CN	AN, ME	CN	1730	1170

2019). Extraímos as características das imagens utilizando três arquiteturas diferentes de redes pré-treinadas: Xception, InceptionResNetV2 e DenseNet201.

- Xception: (CHOLLET, 2017) propõe uma arquitetura de rede neural convolucional profunda inspirada em Inception. Os módulos iniciais consistem em uma série de convoluções em paralelo com diferentes tamanhos de kernel para extrair características. Em vez do Inception, a arquitetura Xception possui convoluções separáveis em profundidade e 36 camadas convolucionais para extrair características essenciais.
- InceptionResNetV2: (SZEGEDY et al., 2017) mostraram que InceptionResNetV2 acelera significativamente o treinamento de redes Inception com a ajuda de conexões residuais. InceptionResNetV2 introduziu uma simplificação significativa nos blocos de inicialização. É uma variação do modelo InceptionV3, que empresta algumas ideias dos modelos ResNet. As conexões residuais permitem o treinamento de redes neurais muito mais profundas, levando a um melhor desempenho.
- DenseNet201: (HUANG et al., 2017) introduziu a Rede Convolucional Densa (DenseNet), que conecta cada camada a todas as outras camadas de forma feed-forward. DenseNet201 é um modelo complexo com 201 camadas.

5.2.1.4 Engenharia de recursos

A análise de componentes principais (PCA) é um método bem conhecido para redução de características. Ao calcular os autovetores da matriz de covariância das entradas originais, o PCA transforma linearmente um vetor de entrada de alta dimensão em um vetor de baixa dimensão cujos componentes não são correlacionados (CAO; CHONG, 2002). O PCA não linear também foi desenvolvido usando diferentes algoritmos. A análise de componentes principais do kernel (KPCA) é um tipo de PCA não linear desenvolvido generalizando o método do kernel em PCA (SCHÖLKOPF; SMOLA; MÜLLER, 1998). Especificamente, o KPCA primeiro mapeia as entradas originais em um

espaço de características de alta dimensão usando o método kernel e, em seguida, calcula o PCA no espaço de características de alta dimensão. O PCA linear no espaço de características de alta dimensão corresponde a um PCA não linear no espaço de entrada original. Outro método de transformação linear, Análise de Componentes Independentes (ICA), tem sido utilizado (HYVARINEN, 1999). Em vez de transformar componentes não correlacionados, o ICA tenta obter componentes estatisticamente independentes nos vetores transformados. O ICA foi inicialmente desenvolvido para separação cega de fontes. Mais tarde, foi generalizado para redução de características.

5.2.1.5 Classificação

Após extrair as características de todas as imagens utilizando modelos CNN, o trabalho proposto utiliza SVM para gerar um modelo para aprender como classificar lesões cutâneas. Esta etapa é chamada de etapa de treinamento. Na etapa de teste, usamos as amostras não vistas para validar o modelo e comparar os resultados previstos com os resultados reais. O problema CLC foi formulado como um problema de classificação binária, então, agora podemos aplicar o algoritmo DiagSVM proposto no Capítulo 4. Também aplicamos o Libsvm (CHANG; LIN, 2011) no problema CLC proposto para comparar os dois resultados.

5.2.2 Resultados para a base de dados ISIC2017

Conforme mencionado anteriormente, este conjunto de dados disponível publicamente contém um total de 2.000 imagens JPG de treinamento, incluindo 374 para a classe ME, 254 para a classe SK e 1372 para a classe BN, e 600 imagens de teste, 117 pertencentes à classe ME, 90 à classe SK, e 393 para a classe BN. Para equilibrar o conjunto de dados, atualizamos o número de amostras de treinamento usando aumento. A tabela 12 mostra o número de amostras para cada classe após o aumento.

Nas tabelas 13, 14 e 15, apresentamos os resultados para cada combinação de extrator CNN e redução de características usando DiagSVM, bem como Libsvm.

A tabela 13 mostra que os algoritmos DiagSVM-PCA, DiagSVM-KPCA e Libsvm-KPCA apresentaram a mesma e melhor métrica de acurácia para o problema "ME, BN - SK" usando Xception como extrator de características. Destacamos os melhores resultados para cada modelo CNN em cada problema. No problema "ME, SK - BN", o DiagSVM-PCA obteve a melhor sensibilidade. Porém, o Libsvm-PCA apresentou melhor acurácia e especificidade. Nas colunas C e σ , apresentamos os parâmetros do SVM, e na coluna N_c , o número de componentes utilizados no algoritmo de redução de características. Os valores dos parâmetros C , σ e N_c foram obtidos a partir do método de busca em grade. Nas tabelas 13, 14 e 15, os algoritmos DiagSVM e Libsvm não utilizam redução de características, portanto, a coluna N_c não é preenchida.

Tabela 13 – Resultados da classificação ISIC2017 usando Xception como extrator de características.

Algoritmo	Problema "ME, BN - SK"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+0	1E-4	-	85.33	91.05	51.16
DiagSVM-PCA	1E+1	1E-4	50	86.83	88.83	62.22
DiagSVM-KPCA	1E+1	1E-1	500	86.83	90.74	57.75
DiagSVM-ICA	1E+0	1E-3	100	85.5	89.83	52.17
Libsvm	1E+1	1E-4	-	86.0	92.43	53.06
Libsvm-PCA	1E+0	1E-4	50	86.17	87.79	60.0
Libsvm-KPCA	1E+1	1E-1	500	86.83	90.74	57.75
Libsvm-ICA	1E+0	1E-4	100	86.17	90.51	54.79

Algoritmo	Problema "ME, SK - BN"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+0	1E-4	-	75.0	63.9	80.76
DiagSVM-PCA	1E+1	1E-4	250	75.17	62.29	83.52
DiagSVM-KPCA	1E+1	1E-1	250	75.0	65.08	79.56
DiagSVM-ICA	1E+0	1E-3	250	73.17	60.18	81.02
Libsvm	1E+1	1E-4	-	76.5	64.47	83.87
Libsvm-PCA	1E+0	1E-4	1000	75.5	65.46	80.3
Libsvm-KPCA	1E+1	1E-1	250	74.67	64.71	79.18
Libsvm-ICA	1E+0	1E-4	250	75.0	73.17	75.47

Na Tabela 14, o algoritmo DiagSVM com redução de características KPCA e ICA apresentou melhor precisão e especificidade, respectivamente, para o problema "ME, BN - SK" utilizando InceptionResNetV2. Para o problema "ME, SK - BN", o algoritmo Libsvm-ICA apresentou melhor precisão do que todos os outros algoritmos. Os resultados obtidos com os modelos InceptionResNetV2 foram melhores que os obtidos com o Xception.

Entre todos os modelos, DenseNet201 apresentou os melhores resultados para o conjunto de dados ISIC2017, conforme mostrado na Tabela 15. Para o problema "ME, BN - SK", o Libsvm-KPCA obteve uma pequena vantagem em termos de precisão. Porém, o Libsvm-KPCA apresentou melhor precisão para o problema "ME, BN - SK".

Com base nos resultados apresentados nas Tabelas 13, 14 e 15, a Figura 15 exibe os resultados dos algoritmos para cada modelo. Escolhemos a métrica acurácia para representar o algoritmo que obteve os melhores resultados. No conjunto de dados ISIC2017, para o problema "ME, BN - SK", o modelo DenseNet201 com algoritmo Libsvm-KPCA obteve os melhores resultados. Para o problema "ME, SK - BN", o modelo DenseNet201 com DiagSVM-KPCA teve melhor desempenho que todos os outros algoritmos. Em geral, DiagSVM e Libsvm tiveram desempenho muito semelhante, indicando que o DiagSVM é uma boa alternativa para ser usado no treinamento de SVM

Tabela 14 – Resultados da classificação ISIC2017 usando InceptionResNetV2 como extrator de características.

Algoritmo	Problema "ME, BN - SK"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+1	1E-4	-	86.17	92.79	53.47
DiagSVM-pca	1E+1	1E-4	50	86.5	87.43	66.67
DiagSVM-kpca	1E+2	1E-1	1000	88.17	90.57	66.1
DiagSVM-ica	1E+3	1E-3	100	85.17	92.18	50.5
Libsvm	1E+1	1E-4	-	86.0	92.43	53.06
Libsvm-pca	1E+1	1E-4	50	86.5	87.43	66.67
Libsvm-kpca	1E+2	1E-1	1000	88.0	90.41	65.52
Libsvm-ica	1E+1	1E-3	100	85.0	85.0	00.00

Algoritmo	Problema "ME, SK - BN"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+1	1E-4	-	76.33	63.6	84.76
DiagSVM-PCA	1E+1	1E-4	1000	76.17	63.11	85.11
DiagSVM-KPCA	1E+2	1E-1	500	76.0	68.86	78.75
DiagSVM-ICA	1E+3	1E-3	250	75.0	62.34	82.93
Libsvm	1E+1	1E-4	-	76.5	64.47	83.87
Libsvm-PCA	1E+1	1E-4	500	76.67	64.63	84.1
Libsvm-KPCA	1E+2	1E-1	250	76.0	66.49	80.44
Libsvm-ICA	1E+1	1E-3	250	77.33	65.24	85.01

em tais problemas.

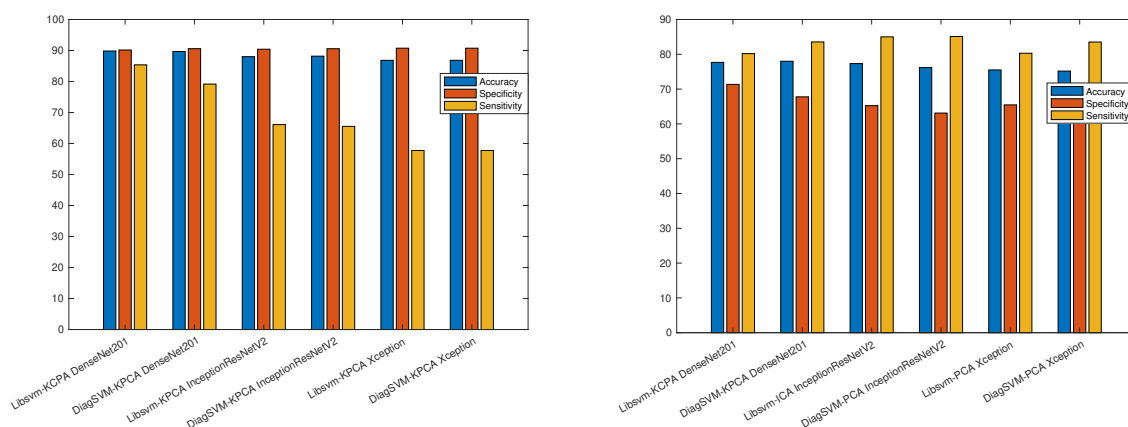


Figura 15 – Representação da acurácia, sensibilidade e especificidade dos algoritmos em cada modelo aplicado a base de dados ISIC2017

5.2.3 Resultado para a base de dados PH2

A base de dados PH2 contém um total de 200 imagens, onde 80 são CN, 80 são AN e 40 são ME. Na Tabela 11, descrevemos a composição da classe após dividir os

Tabela 15 – Resultados da classificação ISIC2017 usando DenseNet201 como extrator de características.

Algoritmo	Problema "ME, BN - SK"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+1	1E-3	-	88.67	92.34	64.1
DiagSVM-pca	1E+1	1E-3	1000	88.5	92.16	63.64
DiagSVM-kpca	1E+2	1E-1	100	89.67	90.58	79.17
DiagSVM-ica	1E+1	1E-1	50	88.83	90.05	74.47
Libsvm	1E+1	1E-3	-	88.67	92.34	64.1
Libsvm-pca	1E+1	1E-3	500	88.5	92.65	62.65
Libsvm-kpca	1E+2	1E-1	500	89.83	90.16	85.37
Libsvm-ica	1E+1	1E-1	50	89.0	89.93	77.27

Algoritmo	Problema "ME, SK - BN"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+1	1E-3	-	75.33	62.24	84.12
DiagSVM-PCA	1E+1	1E-3	1000	75.33	61.94	84.7
DiagSVM-KPCA	1E+2	1E-1	250	78.0	67.77	83.55
DiagSVM-ICA	1E+1	1E-1	50	75.83	63.96	82.8
Libsvm	1E+1	1E-3	-	75.33	62.24	84.12
Libsvm-PCA	1E+1	1E-3	1000	75.5	62.2	84.75
Libsvm-KPCA	1E+2	1E-1	250	77.67	71.35	80.19
Libsvm-ICA	1E+1	1E-1	50	75.83	63.96	82.8

dados em conjuntos de dados de treinamento e teste. Na Tabela 12, apresentamos as amostras de treinamento após o aumento de dados, que usamos para realizar os experimentos nesta seção.

A partir do conjunto de dados PH2, construímos dois problemas de classificação binária, "CN, AN - ML" e "AN, ML - CN". Nas Tabelas 16, 17 e 18, apresentamos os resultados do experimento de todos os algoritmos combinando cada um com um modelo CNN.

Na Tabela 16, utilizando o modelo Xception, para o problema "CN, AN - ML", DiagSVM-ICA e Libsvm-ICA obtiveram a mesma acurácia e especificidade, e DiagSVM-KPCA e Libsvm-KPCA a mesma sensibilidade. Já para o problema "AN, ML - CN", o Libsvm-ICA apresentou a melhor acurácia, enquanto o DiagSVM-ICA apresentou a melhor especificidade.

A partir da Tabela 17, utilizando o modelo InceptionResNetV2, o DiagSVM-PCA obteve os melhores resultados de acurácia e especificidade. Para o problema "AN, ML - CN", o Libsvm-PCA apresentou a melhor acurácia, especificidade e sensibilidade. Os resultados obtidos após usar InceptionResNetV2 como extrator de características foram melhores do que aqueles usando Xception.

Tabela 16 – Resultados da classificação PH2 usando Xception como extrator de características.

Algoritmo	Problema CN, AN - ML"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+1	1E-4	-	88.33	90.2	77.78
DiagSVM-pca	1E+2	1E-4	500	90.0	92.0	80.0
DiagSVM-kpca	1E+3	1E-2	1000	88.33	87.27	100.0
DiagSVM-ica	1E+1	1E-4	50	91.67	93.88	81.82
Libsvm	1E+2	1E-4	2048	90.0	90.38	87.5
Libsvm-pca	1E+0	1E-4	100	90.0	92.0	94.44
Libsvm-kpca	1E+3	1E-2	1000	88.33	87.27	100.0
Libsvm-ica	1E+2	1E-3	50	91.67	93.88	94.44
Algoritmo	Problema "AN, ML - CN"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+1	1E-4	-	73.33	77.78	66.67
DiagSVM-PCA	1E+2	1E-4	50	80.0	81.58	77.27
DiagSVM-KPCA	1E+3	1E-2	100	80.0	83.33	75.0
DiagSVM-ICA	1E+1	1E-4	50	76.67	86.67	66.67
Libsvm	1E+2	1E-4	2048	73.33	79.41	65.38
Libsvm-PCA	1E+0	1E-4	50	80.0	81.58	77.27
Libsvm-KPCA	1E+3	1E-2	100	78.33	79.49	76.19
Libsvm-ICA	1E+2	1E-3	100	81.67	85.71	76.0

A Tabela 18 mostra que Libsvm e DiagSVM usando PCA e ICA tiveram desempenho idêntico para o problema "CN, AN - ML" usando o modelo DenseNet201. Para o problema "AN, ML - CN", DiagSVM-KPCA e Libsvm-KPCA apresentaram o mesmo resultado em acurácia.

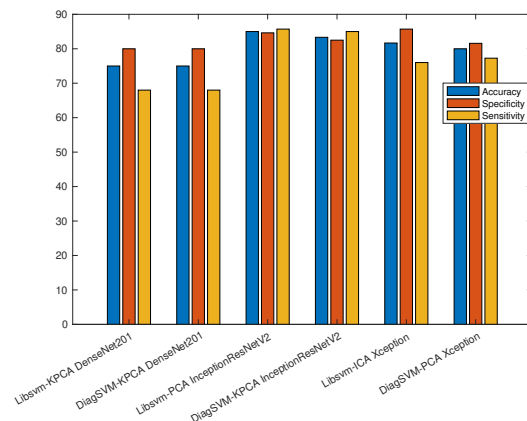
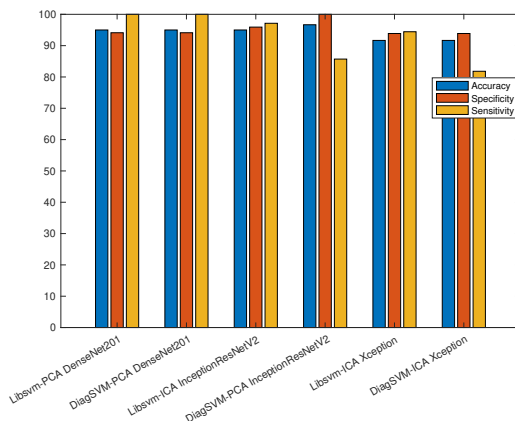
Dentre todos os algoritmos aplicados ao problema "CN, AN - ML", o DiagSVM-PCA foi o que apresentou melhor desempenho. E para o problema "AN, ML - CN", o Libsvm-PCA obteve os melhores resultados, ambos utilizando o modelo InceptionResNetV2 como extrator de características.

De forma semelhante ao gráfico apresentado na Seção 5.2.2, apresentamos na Figura 16 os resultados das métricas dos melhores algoritmos para cada problema e modelo. Os resultados do Libsvm e do DiagSVM para o problema "CN, AN - ML" foram semelhantes utilizando os modelos Xception e DenseNet201, mas para o modelo InceptionResNetV2 o DiagSVM obteve o melhor resultado comparado a todos os outros algoritmos e modelos.

Tabela 17 – Resultados da classificação PH2 usando InceptionResNetV2 como extrator de características.

Algoritmo	Problema CN, AN - ML"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+2	1E-4	1536	93.33	95.83	83.33
DiagSVM-pca	1E+0	1E-4	50	96.67	100.0	85.71
DiagSVM-kpca	1E+2	1E-2	500	91.67	92.16	88.89
DiagSVM-ica	1E+3	1E-4	50	95.0	95.92	90.91
Libsvm	1E+2	1E-4	-	93.33	95.83	83.33
Libsvm-pca	1E+1	1E-4	50	91.67	95.74	91.89
Libsvm-kpca	1E+2	1E-2	250	90.0	90.38	97.14
Libsvm-ica	1E+1	1E-3	50	95.0	95.92	97.14

	Problema "AN, ML - CN"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+2	1E-4	1536	83.33	84.21	81.82
DiagSVM-PCA	1E+0	1E-4	50	83.33	82.5	85.0
DiagSVM-KPCA	1E+2	1E-2	1000	83.33	82.5	85.0
DiagSVM-ICA	1E+3	1E-4	100	83.33	82.5	85.0
Libsvm	1E+2	1E-4	-	81.67	80.49	84.21
Libsvm-PCA	1E+1	1E-4	250	85.0	84.62	85.71
Libsvm-KPCA	1E+2	1E-2	1000	83.33	82.5	85.0
Libsvm-ICA	1E+1	1E-3	100	83.33	82.5	85.0



(a) Resultados para o problema "CN, AN - ML". (b) Resultados para o problema "AN, ML - CN"

Figura 16 – Representação da acurácias, sensibilidade e especificidade dos melhores algoritmos em cada modelo aplicado a base de dados PH2.

5.3 Considerações Finais do Capítulo

Foi apresentado, neste capítulo, experimentos numéricos envolvendo a proposta da tese. Para isso, considerou-se analisar a performance da proposta com a base de dados de benchmark popularmente conhecidos na literatura. Além disso, também foi

Tabela 18 – Resultados da classificação PH2 usando DenseNet201 como extrator de recursos.

Algoritmo	Problema CN, AN - ML"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+3	1E-4	-	93.33	92.31	100.0
DiagSVM-pca	1E+3	1E-4	1000	95.0	94.12	100.0
DiagSVM-kpca	1E+2	1E-1	100	93.33	92.31	100.0
DiagSVM-ica	1E+1	1E-4	50	95.0	94.12	100.0
Libsvm	1E+0	1E-4	1920	95.0	94.12	100.0
Libsvm-pca	1E+0	1E-4	50	95.0	94.12	100.0
Libsvm-kpca	1E+2	1E-1	100	93.33	92.31	100.0
Libsvm-ica	1E+1	1E-2	50	95.0	94.12	100.0
	Problema "AN, ML - CN"					
	C	σ	Nc	A_c	S_p	S_e
DiagSVM	1E+3	1E-4	-	66.67	73.53	57.69
DiagSVM-PCA	1E+3	1E-4	250	68.33	79.31	58.06
DiagSVM-KPCA	1E+2	1E-1	50	75.0	80.0	68.0
DiagSVM-ICA	1E+1	1E-4	100	71.67	82.76	61.29
Libsvm	1E+0	1E-4	1920	66.67	73.53	57.69
Libsvm-PCA	1E+0	1E-4	50	70.0	78.12	60.71
Libsvm-KPCA	1E+2	1E-1	50	75.0	80.0	68.0
Libsvm-ICA	1E+1	1E-2	50	73.33	83.33	63.33

comparado algumas métricas de performance com métodos do estado da arte para o problema tratado, sendo eles a acurácia, sensibilidade e especificidade.

Os estudos relativos a comparação da performance da proposta com os algoritmos da literatura ilustram também que os conjuntos de dados escalados dão destaque para a proposta. Para um problema de treinamento SVM, o *framework* DiagSVM proposto apresenta um desempenho competitivo comparado ao algoritmo Libsvm. Algumas sugestões de continuidade da pesquisa são dadas no próximo capítulo.

6 Conclusão

Este trabalho apresentou um novo solucionador para o problema de treinamento do SVM não linear. O solucionador trata-se de um *framework* denominado DiagSVM que consiste em uma combinação de duas abordagens. No primeiro estágio, é utilizada uma técnica de aproximação quadrática separável da função objetivo do problema, resultando em um novo subproblema caracterizado como um problema da mochila quadrática separável (PMQS). De acordo com nossas pesquisas na literatura, tal abordagem ainda não foi utilizada para resolver um problema de otimização não linear no formato que o treinamento de SVM dual possui. Os resultados numéricos também indicam que o desempenho e a performance da proposta são satisfatórios em comparação com o solucionador Libsvm para os conjuntos de dados testados, satisfazendo os objetivos apresentados na introdução deste trabalho.

Ainda relacionado aos objetivos do trabalho, o *framework* DiagSVM mostrou-se flexível com relação às possibilidades de implementações no primeiro e no segundo estágio para resolução do problema de treinamento de SVM, onde um conjunto de possibilidades foi proposto e testado.

Em nossos estudos preliminares, também desenvolvemos uma estratégia de ponto fixo (ALVES et al., 2023) baseada em Münnich, Sachs e Wagner (2012) e suas referências, que forneceu ao *framework* mais competitividade comparado a outros métodos de busca por um zero de uma função.

Nos testes numéricos, a aproximação esférica e a aproximação quasi-Cauchy, detalhadas na Seção 4.1, foram utilizadas na primeira etapa do *framework*. O *framework* usa métodos de busca por um zero de uma função no segundo estágio, detalhado na Seção 4.2. Nos resultados, podemos observar a estabilidade do algoritmo com a grande maioria das combinações devido às propriedades dos subproblemas gerados.

Implementamos a proposta em MATLAB e Python e aplicamos a vários problemas clássicos da literatura. No geral, em termos de acurácia, sensibilidade e especificidade, a proposta apresentou resultados satisfatórios em comparação com um dos métodos mais popular da literatura: Libsvm.

6.1 Contribuições

Destacam-se como principais contribuições desse trabalho:

- O solucionador SVM proposto (DiagSVM) permite o uso de métodos diversos tanto na etapa de aproximação diagonal quanto na resolução dos subproblemas

aproximados.

- O DiagSVM combina várias alternativas para encontrar bons resultados de treinamento nos problemas de SVM.
- O DiagSVM possibilita o treinamento de volumes médios de dados, mantendo as métricas competitivas com o método de estado da arte Libsvm.
- A comparação entre PCA, KPCA e ICA mostra que a redução de características melhora o desempenho do SVM aplicado à classificação de lesões cutâneas (CLC).
- O estudo comparou três arquiteturas de redes neurais convolucionais pré-treinadas (Xception, InceptionResNetV2 e DenseNet201) como extratoras de características para melhorar a taxa de acurácia da classificação.
- Os testes realizados com os conjuntos de dados ISIC2017 e PH2 mostraram que o DiagSVM é eficiente na classificação binária de lesões cutâneas, obtendo resultados favoráveis comparado a métodos existentes.

6.2 Trabalhos Futuros

Sugestões para novas pesquisas são listadas abaixo.

- Expandir o estudo de métodos de aproximação quadrática separáveis para aplicar no primeiro estágio do *framework*.
- Expandir o estudo de métodos de busca por um zero de uma função para aplicar no segundo estágio do *framework*.
- Acelerar a convergência do *framework* utilizando novos meios de atualizar a matriz diagonal de acordo com os estudos de [Zhu, Nazareth e Wolkowicz \(1997\)](#), de modo a melhorar o tempo computacional mantendo boa acurácia na classificação de SVM.
- Desenvolver uma versão do DiagSVM otimizada para execução em GPUs, utilizando frameworks como CUDA ou OpenCL, para acelerar o processo de treinamento e classificação.
- Desenvolver e otimizar implementações de diferentes kernels SVM (e.g., RBF, polinomial) para execução eficiente em GPUs, focando na redução do tempo de computação e na utilização eficiente da memória.
- Implementar e avaliar algoritmos de busca por um zero de uma função paralelizados, utilizados no DiagSVM, para acelerar a resolução de subproblemas de otimização em GPUs.

6.3 Produções Científicas

A Tabela 19 apresenta os artigos diretamente relacionados ao método proposto, DiagSVM. Além disso, a Tabela 20 lista os artigos científicos publicados e submetidos em outras aplicações relacionadas a otimização desde o início do doutorado.

Tabela 19 – Produções científicas em relação ao método proposto para classificação de lesões cutâneas.

Artigo	Tipo	Scopus	Status
A new SVM solver applied to skin lesion classification. Em: Statistics, Optimization & Information Computing. (SILVA et al., 2024)	Periódico	59º percentil	Publicado

Tabela 20 – Produções científicas em outras aplicações relacionadas a otimização matemática

Artigo	Tipo	Scopus	Status
An accelerated fixed-point algorithm applied to quadratic convex separable knapsack problems. Em: Journal of Control Science and Engineering. (ALVES et al., 2023)	Periódico	42º percentil	Publicado

Referências

- AITKEN, A. On bernoulli's numerical solution of algebraic equations.-proc. roy. soc., edinburgh, ser. a, 46. 1925.
- ALIZADEH, S. M.; MAHLOOJIFAR, A. Automatic skin cancer detection in dermoscopy images by combining convolutional neural networks and texture features. **International Journal of Imaging Systems and Technology**, Wiley Online Library, v. 31, n. 2, p. 695–707, 2021.
- ALQURAN, H.; QASMIEH, I. A.; ALQUDAH, A. M.; ALHAMMOURI, S.; ALAWNEH, E.; ABUGHAZALEH, A.; HASAYEN, F. The melanoma skin cancer detection and classification using support vector machine. In: IEEE. **2017 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)**. [S.l.], 2017. p. 1–5.
- ALVES, A.; SILVA, J.; SANTOS, P. Methods based on approximation of the max function applied to the separable quadratic knapsack problem. **Unpublished manuscript**, 2023.
- ALVES, A.; SILVA, J.; SANTOS, P.; MATIOLI, L.; SOUZA, S. An accelerated fixed-point algorithm applied to quadratic convex separable knapsack problems. **Journal of Control Science and Engineering**, 2023.
- ANDERSON, D. G. Iterative procedures for nonlinear integral equations. **Journal of the ACM (JACM)**, ACM New York, NY, USA, v. 12, n. 4, p. 547–560, 1965.
- ANDREI, N. A diagonal quasi-newton updating method for unconstrained optimization. **Numerical Algorithms**, Springer, v. 81, n. 2, p. 575–590, 2019.
- ANDREI, N. Diagonal approximation of the hessian by finite differences for unconstrained optimization. **Journal of Optimization Theory and Applications**, Springer, v. 185, n. 3, p. 859–879, 2020.
- ANDREI, N. A new accelerated diagonal quasi-newton updating method with scaled forward finite differences directional derivative for unconstrained optimization. **Optimization**, Taylor & Francis, v. 70, n. 2, p. 345–360, 2021.
- BALDOMINOS, A.; SAEZ, Y.; ISASI, P. A survey of handwritten character recognition with mnist and emnist. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 9, n. 15, p. 3169, 2019.
- BAUMANN, S.; KLYMAK, M. Fixedpoint: A suite of acceleration algorithms with applications. 2021.
- BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear programming: theory and algorithms**. [S.l.]: John Wiley & Sons, 2013.
- BHAVSAR, H.; PANCHAL, M. H. A review on support vector machine for data classification. **International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)**, v. 1, n. 10, p. 185–189, 2012.

- BIRGIN, E. G.; MARTÍNEZ, J. M.; RAYDAN, M. Spectral projected gradient methods: review and perspectives. **Journal of Statistical Software**, v. 60, p. 1–21, 2014.
- BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.: s.n.], 1992. p. 144–152.
- BRETTTHAUER, K. M.; SHETTY, B. The nonlinear knapsack problem—algorithms and applications. **European Journal of Operational Research**, Elsevier, v. 138, n. 3, p. 459–472, 2002.
- CAO, L.; CHONG, W. Feature extraction in support vector machine: a comparison of pca, xpca and ica. In: IEEE. **Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02**. [S.l.], 2002. v. 2, p. 1001–1005.
- CASSIDY, B.; KENDRICK, C.; BRODZICKI, A.; JAWOREK-KORJAKOWSKA, J.; YAP, M. H. Analysis of the isic image datasets: Usage, benchmarks and recommendations. **Medical image analysis**, Elsevier, v. 75, p. 102305, 2022.
- CERVANTES, J.; GARCIA-LAMONT, F.; RODRÍGUEZ-MAZAHUA, L.; LOPEZ, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. **Neurocomputing**, Elsevier, v. 408, p. 189–215, 2020.
- CHANG, C.-C.; LIN, C.-J. Libsvm: a library for support vector machines. **ACM transactions on intelligent systems and technology (TIST)**, Acm New York, NY, USA, v. 2, n. 3, p. 1–27, 2011.
- CHAUHAN, V. K.; DAHIYA, K.; SHARMA, A. Problem formulations and solvers in linear svm: a review. **Artificial Intelligence Review**, Springer, v. 52, n. 2, p. 803–855, 2019.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1251–1258.
- CODELLA, N. C.; GUTMAN, D.; CELEBI, M. E.; HELBA, B.; MARCHETTI, M. A.; DUSZA, S. W.; KALLOO, A.; LIOPYRIS, K.; MISHRA, N.; KITTLER, H. et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: IEEE. **2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)**. [S.l.], 2018. p. 168–172.
- COMINETTI, R.; MASCARENHAS, W. F.; SILVA, P. J. A newton's method for the continuous quadratic knapsack problem. **Mathematical Programming Computation**, Springer, v. 6, n. 2, p. 151–169, 2014.
- CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.
- CRISTIANINI, N.; SHAW-TAYLOR, J. et al. **An introduction to support vector machines and other kernel-based learning methods**. [S.l.]: Cambridge university press, 2000.

- DAI, Y.-H.; FLETCHER, R. New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. **Mathematical Programming**, Springer, v. 106, n. 3, p. 403–421, 2006.
- DAVIDON, W. C. Variable metric method for minimization. 5 1959.
- DENNIS JR, J. E.; WOLKOWICZ, H. Sizing and least-change secant methods. **SIAM Journal on Numerical Analysis**, SIAM, v. 30, n. 5, p. 1291–1314, 1993.
- DUGAS, C.; BENGIO, Y.; BÉLISLE, F.; NADEAU, C.; GARCIA, R. Incorporating second-order functional knowledge for better option pricing. **Advances in neural information processing systems**, v. 13, 2000.
- ELLIOTT, D. L. **A better activation function for artificial neural networks**. [S.l.], 1993.
- ETMAN, L.; GROENWOLD, A. A.; ROODA, J. First-order sequential convex programming using approximate diagonal qp subproblems. **Structural and Multidisciplinary Optimization**, Springer, v. 45, p. 479–488, 2012.
- FAN, R.-E.; CHANG, K.-W.; HSIEH, C.-J.; WANG, X.-R.; LIN, C.-J. Liblinear: A library for large linear classification. **the Journal of machine Learning research**, JMLR. org, v. 9, p. 1871–1874, 2008.
- FLETCHER, R.; REEVES, C. M. Function minimization by conjugate gradients. **The computer journal**, Oxford University Press, v. 7, n. 2, p. 149–154, 1964.
- GHOLAMI, R.; FAKHARI, N. Support vector machine: principles, parameters, and applications. In: **Handbook of neural computation**. [S.l.]: Elsevier, 2017. p. 515–535.
- GOLDFELD, S. M.; QUANDT, R. E.; TROTTER, H. F. Maximization by quadratic hill-climbing. **Econometrica: Journal of the Econometric Society**, JSTOR, p. 541–551, 1966.
- GUO, Y.; LIU, Y.; OERLEMANS, A.; LAO, S.; WU, S.; LEW, M. S. Deep learning for visual understanding: A review. **Neurocomputing**, Elsevier, v. 187, p. 27–48, 2016.
- HASAN, M. K.; AHAMAD, M. A.; YAP, C. H.; YANG, G. A survey, review, and future trends of skin lesion segmentation and classification. **Computers in Biology and Medicine**, Elsevier, v. 155, p. 106624, 2023.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Springer, 2009. v. 2.
- HSIEH, C.-J.; CHANG, K.-W.; LIN, C.-J.; KEERTHI, S. S.; SUNDARARAJAN, S. A dual coordinate descent method for large-scale linear svm. In: **Proceedings of the 25th international conference on Machine learning**. [S.l.: s.n.], 2008. p. 408–415.
- HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. et al. **A practical guide to support vector classification**. [S.l.]: Taipei, Taiwan, 2003.
- HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 4700–4708.

- HYVARINEN, A. Fast and robust fixed-point algorithms for independent component analysis. **IEEE transactions on Neural Networks**, IEEE, v. 10, n. 3, p. 626–634, 1999.
- JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.
- JOACHIMS, T. 11 making large-scale support vector machine learning practical. In: **Advances in kernel methods: support vector learning**. [S.l.]: MIT press, 1999. p. 169.
- JOACHIMS, T. Training linear svms in linear time. In: **Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2006. p. 217–226.
- KASSANI, S. H.; KASSANI, P. H. A comparative study of deep learning architectures on melanoma detection. **Tissue and Cell**, Elsevier, v. 58, p. 76–83, 2019.
- KHAN, M. A.; JAVED, M. Y.; SHARIF, M.; SABA, T.; REHMAN, A. Multi-model deep neural network based features extraction and optimal selection approach for skin lesion classification. In: IEEE. **2019 international conference on computer and information sciences (ICCIS)**. [S.l.], 2019. p. 1–7.
- KIWIEL, K. Variable fixing algorithms for the continuous quadratic knapsack problem. **Journal of Optimization Theory and Applications**, Springer, v. 136, n. 3, p. 445–458, 2008.
- LECUN, Y.; CORTES, C.; BURGESS, C. **MNIST handwritten digit database**. [S.l.]: Florham Park, NJ, USA, 2010.
- LIN, C.-J.; LUCIDI, S.; PALAGI, L.; RISI, A.; SCIANDRONE, M. Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds. **Journal of Optimization Theory and Applications**, Springer, v. 141, n. 1, p. 107–126, 2009.
- LIN, C.-J.; WENG, R. C.; KEERTHI, S. S. Trust region newton methods for large-scale logistic regression. In: **Proceedings of the 24th international conference on Machine learning**. [S.l.: s.n.], 2007. p. 561–568.
- LIU, P.; CHOO, K.-K. R.; WANG, L.; HUANG, F. Svm or deep learning? a comparative study on remote sensing image classification. **Soft Computing**, Springer, v. 21, n. 23, p. 7053–7065, 2017.
- LORENA, A. C.; CARVALHO, A. C. D. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007.
- LUCIDI, S.; PALAGI, L.; RISI, A.; SCIANDRONE, M. A convergent hybrid decomposition algorithm model for svm training. **IEEE transactions on neural networks**, IEEE, v. 20, n. 6, p. 1055–1060, 2009.
- MAHBOD, A.; SCHAEFER, G.; WANG, C.; ECKER, R.; ELLINGE, I. Skin lesion classification using hybrid deep neural networks. In: IEEE. **ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)**. [S.l.], 2019. p. 1229–1233.

MENDONÇA, T.; FERREIRA, P. M.; MARQUES, J. S.; MARCAL, A. R.; ROZEIRA, J. Ph 2-a dermoscopic image database for research and benchmarking. In: IEEE. **2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)**. [S.l.], 2013. p. 5437–5440.

MERCER, J. Xvi. functions of positive and negative type, and their connection the theory of integral equations. **Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character**, The Royal Society London, v. 209, n. 441-458, p. 415–446, 1909.

MULLER, K.-R.; MIKA, S.; RATSCH, G.; TSUDA, K.; SCHOLKOPF, B. An introduction to kernel-based learning algorithms. **IEEE transactions on neural networks**, IEEE, v. 12, n. 2, p. 181–201, 2001.

MÜNNICH, R. T.; SACHS, E. W.; WAGNER, M. Numerical solution of optimal allocation problems in stratified sampling under box constraints. **AStA Advances in Statistical Analysis**, Springer, v. 96, n. 3, p. 435–450, 2012.

PICCIALLI, V.; SCIANDRONE, M. Nonlinear optimization and support vector machines. **Annals of Operations Research**, Springer, p. 1–33, 2022.

PLATT, J. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

QIU, J.; WU, Q.; DING, G.; XU, Y.; FENG, S. A survey of machine learning for big data processing. **EURASIP Journal on Advances in Signal Processing**, Springer, v. 2016, n. 1, p. 1–16, 2016.

QIU, S.; LANE, T. Parallel computation of rbf kernels for support vector classifiers. In: SIAM. **Proceedings of the 2005 SIAM International Conference on Data Mining**. [S.l.], 2005. p. 334–345.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015.

SCHÖLKOPF, B.; SMOLA, A.; MÜLLER, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. **Neural computation**, MIT Press, v. 10, n. 5, p. 1299–1319, 1998.

SERAFINI, T.; ZANGHIRATI, G.; ZANNI, L. Gradient projection methods for quadratic programs and applications in training support vector machines. **Optimization Methods and Software**, Taylor & Francis, v. 20, n. 2-3, p. 353–378, 2005.

SHALEV-SHWARTZ, S.; SINGER, Y.; SREBRO, N. Pegasos: Primal estimated sub-gradient solver for svm. In: **Proceedings of the 24th international conference on Machine learning**. [S.l.: s.n.], 2007. p. 807–814.

SHETTY, B.; FERNANDES, R.; RODRIGUES, A. P.; CHENGODEN, R.; BHATTACHARYA, S.; LAKSHMANNA, K. Skin lesion classification of dermoscopic images using machine learning and convolutional neural network. **Scientific Reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 18134, 2022.

SILVA, J. O. L.; ALVES, A.; SANTOS, P. S. M.; MATIOLI, L. C. A new svm solver applied to skin lesion classification. **Statistics, Optimization & Information Computing**, 2024. Disponível em: <<https://api.semanticscholar.org/CorpusID:270353921>>.

SIVAPALAN, G.; NUNDY, K. K.; DEV, S.; CARDIFF, B.; JOHN, D. Annet: A lightweight neural network for ecg anomaly detection in iot edge sensors. **IEEE Transactions on Biomedical Circuits and Systems**, IEEE, v. 16, n. 1, p. 24–35, 2022.

SNYMAN, J.; HAY, A. The spherical quadratic steepest descent (sqsd) method for unconstrained minimization with no explicit line searches. **Computers & Mathematics with Applications**, Elsevier, v. 42, n. 1-2, p. 169–178, 2001.

SNYMAN, J. A.; WILKE, D. N. et al. **Practical mathematical optimization**. [S.l.]: Springer, 2005.

SORENSEN, D. C. Newton's method with a model trust region modification. **SIAM Journal on Numerical Analysis**, SIAM, v. 19, n. 2, p. 409–426, 1982.

SUYKENS, J. A.; VANDEWALLE, J. Least squares support vector machine classifiers. **Neural processing letters**, Springer, v. 9, n. 3, p. 293–300, 1999.

SVANBERG, K. A class of globally convergent optimization methods based on conservative convex separable approximations. **SIAM journal on optimization**, SIAM, v. 12, n. 2, p. 555–573, 2002.

SVENSÉN, M.; BISHOP, C. M. **Pattern recognition and machine learning**. [S.l.]: Springer Berlin/Heidelberg, Germany, 2007.

SZEGEDY, C.; IOFFE, S.; VANHOUCHE, V.; ALEMI, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Proceedings of the AAAI conference on artificial intelligence**. [S.l.: s.n.], 2017. v. 31, n. 1.

TANG, Y. Deep learning using linear support vector machines. **arXiv preprint arXiv:1306.0239**, 2013.

WALKER, H. F.; NI, P. Anderson acceleration for fixed-point iterations. **SIAM Journal on Numerical Analysis**, SIAM, v. 49, n. 4, p. 1715–1735, 2011.

WANG, Y.; BAEK, J.; TANG, Y.; DU, J.; HILLMAN, M.; CHEN, J.-S. Support vector machine guided reproducing kernel particle method for image-based modeling of microstructures. **Computational Mechanics**, Springer, v. 73, n. 4, p. 907–942, 2024.

WIEMANN, P. F.; KNEIB, T.; HAMBUCKERS, J. Using the softplus function to construct alternative link functions in generalized linear models and beyond. **arXiv preprint arXiv:2111.14207**, 2021.

WILKE, D. N.; KOK, S.; GROENWOLD, A. A. The application of gradient-only optimization methods for problems discretized using non-constant methods. **Structural and Multidisciplinary Optimization**, Springer, v. 40, p. 433–451, 2010.

YOKOTA, R.; BARBA, L. A.; KNEPLEY, M. G. Petrbf—a parallel $O(n)$ algorithm for radial basis function interpolation with gaussians. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 199, n. 25-28, p. 1793–1804, 2010.

YU, Z.; NI, D.; CHEN, S.; QIN, J.; LI, S.; WANG, T.; LEI, B. Hybrid dermoscopy image classification framework based on deep convolutional neural network and fisher vector. In: IEEE. **2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)**. [S.l.], 2017. p. 301–304.

ZANNI, L.; SERAFINI, T.; ZANGHIRATI, G.; BENNETT, K. P.; PARRADO-HERNÁNDEZ, E. Parallel software for training large scale support vector machines on multiprocessor systems. **Journal of Machine Learning Research**, v. 7, n. 7, 2006.

ZHENG, H.; YANG, Z.; LIU, W.; LIANG, J.; LI, Y. Improving deep neural networks using softplus units. In: IEEE. **2015 International joint conference on neural networks (IJCNN)**. [S.l.], 2015. p. 1–4.

ZHU, M.; NAZARETH, J. L.; WOLKOWICZ, H. The quasi-cauchy relation and diagonal updating. **SIAM Journal on Optimization**, SIAM, v. 9, n. 4, p. 1192–1204, 1997.