

UNIVERSIDADE FEDERAL DO PARANÁ
Evelin Heringer Manoel Krulikovski

**ANÁLISE TEÓRICA DE MÁQUINAS DE VETORES SUPORTE E
APLICAÇÃO A CLASSIFICAÇÃO DE CARACTERES**

Curitiba
2017

UNIVERSIDADE FEDERAL DO PARANÁ
Evelin Heringer Manoel Krulikovski

**ANÁLISE TEÓRICA DE MÁQUINAS DE VETORES SUPORTE E
APLICAÇÃO A CLASSIFICAÇÃO DE CARACTERES**

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de Mestre em Matemática.

Orientadora: Profa. Dra. Mael Sachine.

Coorientador: Prof. Dr. Ademir Alves Ribeiro.

Curitiba
2017



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós-Graduação MATEMÁTICA

ATA Nº81

ATA DE SESSÃO PÚBLICA DE DEFESA DE MESTRADO PARA A OBTENÇÃO DO GRAU DE MESTRE EM MATEMÁTICA

No dia trinta e um de Julho de dois mil e dezessete às 09:00 horas, na sala Anfiteatro A, Universidade Federal do Paraná, Centro Politécnico, foram instalados os trabalhos de arguição da mestranda **EVELIN HERINGER MANOEL KRULIKOVSKI** para a Defesa Pública de sua dissertação intitulada **ANÁLISE TEÓRICA DE MÁQUINAS DE VETORES SUPORTE E APLICAÇÃO A CLASSIFICAÇÃO DE CARACTERES**. A Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná, foi constituída pelos seguintes Membros: MAEL SACHINE (UFPR), LUCAS GARCIA PEDROSO (UFPR), PAULO J. SILVA E SILVA (Unicamp). Dando início à sessão, a presidência passou a palavra a discente, para que a mesma expusesse seu trabalho aos presentes. Em seguida, a presidência passou a palavra a cada um dos Examinadores, para suas respectivas arguições. A aluna respondeu a cada um dos arguidores. A presidência retomou a palavra para suas considerações finais. A Banca Examinadora, então, reuniu-se e, após a discussão de suas avaliações, decidiu-se pela APROVAÇÃO da aluna. A mestranda foi convidada a ingressar novamente na sala, bem como os demais assistentes, após o que a presidência fez a leitura do Parecer da Banca Examinadora. A aprovação no rito de defesa deverá ser homologada pelo Colegiado do programa, mediante o atendimento de todas as indicações e correções solicitadas pela banca dentro dos prazos regimentais do programa. A outorga do título de mestre está condicionada ao atendimento de todos os requisitos e prazos determinados no regimento do Programa de Pós-Graduação. Nada mais havendo a tratar a presidência deu por encerrada a sessão, da qual eu, MAEL SACHINE, lavrei a presente ata, que vai assinada por mim e pelos membros da Comissão Examinadora.

Curitiba, 31 de Julho de 2017.


MAEL SACHINE

Presidente da Banca Examinadora (UFPR)


LUCAS GARCIA PEDROSO

Avaliador Interno (UFPR)


PAULO J. SILVA E SILVA

Avaliador Externo (Unicamp)



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós-Graduação MATEMÁTICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **EVELIN HERINGER MANOEL KRULIKOVSKI** intitulada: **ANÁLISE TEÓRICA DE MÁQUINAS DE VETORES SUPORTE E APLICAÇÃO A CLASSIFICAÇÃO DE CARACTERES**, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 31 de Julho de 2017.


MAEL SACHINE

Presidente da Banca Examinadora (UFPR)


LUCAS GARCIA PEDROSO

Avaliador Interno (UFPR)


PAULO J. SILVA E SILVA

Avaliador Externo (Unicamp)

*A Deus.
Ao meu amor Alex,
por estar sempre comigo.*

Agradecimentos

Agradeço primeiramente a Deus, por ter realizado tantas obras em minha vida e pela manifestação de sua vontade.

A minha família, que me apoiou nessa caminhada e os meus avôs por serem as minhas rampas de apoio incondicional.

Ao meu esposo Alex, por ter encontrado nele o mais sublime amor e saber que ele fará parte de todos os momentos da minha vida.

Aos meus orientadores Dra. Mael Sachine e Dr. Ademir Alves Ribeiro, pela amizade que construímos durante a orientação. Minha imensa gratidão, pelos seus conselhos, pela sua disponibilidade, pela excelência na orientação e principalmente pelo carinho. Estou certa de que neste trabalho há muito deles.

Gostaria de expressar os mais sinceros agradecimentos ao Professor Dr. Celso Carnieri pela sua colaboração indispensável, em que apresentou-me a técnica das Máquinas de Vetores Suporte e o problema da leitura de placas automotivas. Além disso, mostrou em nosso encontro excelência e amor profissional.

As professoras Dra. Sandra Augusta Santos e Dra. Tiara Martini por terem sido atenciosas indicando à utilização do Algencan, tendo uma grande intervenção na execução desta dissertação.

Ao Professor Dr. Francisco Nogueira Calmon Sobral, por ter disponibilizado tempo na instalação do Algencan, com interface em Matlab e pela paciência durante todo o processo até a utilização do mesmo, sem ele certamente não teria conseguido. Sem dúvidas, sua dedicação e amor pelo que faz motivou-me a aprender e trabalhar mais no programa e buscar fazer o melhor em pesquisa.

Aos professores Dr. Lucas Garcia Pedroso e Dr. Paulo José da Silva e Silva, por aceitarem participar da banca examinadora e pelos comentários e sugestões que ajudaram muito a melhorar na versão final desta dissertação.

Ao programa de Pós-Graduação em Matemática da UFPR pela oportunidade e formação de qualidade propiciada.

Não posso deixar de agradecer ainda à Cida e Dona Elizia, muito obrigada. A todos que contribuíram de alguma forma, com gestos de carinho e atenção.

Ao apoio financeiro, disponibilizado pela Capes e pela Fundação Araucária.

*“Paz se possível,
mas a verdade a
qualquer custo.”
Martinho Lutero*

Resumo

O objetivo geral deste trabalho foi realizar um estudo teórico e uma pequena aplicação sobre SVM, que inclui relatar justificativas para o uso de tal técnica e exibir sua interpretação geométrica e perspectiva analítica. Para aplicar a técnica em problemas de classificação, buscamos fundamentar matematicamente sua utilização, visto que envolve um problema de programação quadrática, convexa e com restrições. Para a análise da técnica, utilizamos a teoria de dualidade Lagrangiana, que notamos facilitar os cálculos e a análise das soluções. Além disso, reescrevemos resultados que usam ponto de sela, sem precisar deste conceito. Estabelecemos algumas implicações e exibimos alguns contraexemplos, para mostrar que certos resultados decorrentes da técnica SVM encontrados na literatura não são precisos. Foram feitas algumas comparações, para analisar os diferentes parâmetros da função *Kernel* Gaussiana, usada para resolver o problema quando não for possível encontrar uma função de decisão no espaço de entrada. Verificamos que a eficiência da técnica depende da escolha do parâmetro de regularização, da função *Kernel* e seus respectivos parâmetros. Tal técnica foi testada sobre um banco de dados criado artificialmente e composto de imagens de caracteres. Para a implementação computacional, usamos a interface do programa Algencan e algumas funções próprias do Matlab.

Palavras-chave: Máquinas de Vetores Suporte. Programação não linear. Otimização com restrições. Dualidade Lagrangiana. Processamento de imagens. Algencan.

Abstract

The general objective of this work was to perform a theoretical study and a small application about SVM, which includes reporting justifications for the use of such technique and showing its geometric interpretation and analytical perspective. In order to apply the technique to classification problems, we seek to base its use mathematically, since it involves a quadratic, convex and constrained programming problem. For the analysis of the technique, we use the theory of Lagrangian duality, which we noticed to facilitate the calculations and the analysis of the solutions. In addition, we rewrite results using a saddle point, without needing this concept. We have established some implications and have shown some counterexamples to show that certain results from the SVM technique found in the literature are not accurate. Some comparisons have been made to analyze the different parameters of the Gaussian kernel function used to solve the problem when it is not possible to find a decision function in the input space. We have verified that the efficiency of the technique depends on the choice of the regularization parameter, the kernel function and its parameters. This technique was tested on an artificially created database composed of character images. For the computational implementation, we used the Algencan program interface and some of Matlab's own functions.

Keywords: Support Vector Machine. Nonlinear programming. Optimization with constraints. Lagrangian duality. Image processing. Algencan.

Lista de Figuras

1	Conjunto de dados para classificação.	4
1.1	Exemplo de conjunto convexo e não convexo.	14
1.2	Função convexa.	15
2.1	Dados lineares, com margem flexível e não lineares.	28
2.2	Transformação de caractere em vetor.	29
2.3	Hiperplanos.	31
2.4	Separação de dados.	36
2.5	Dados sobre os hiperplanos tracejados correspondem as restrições ativas.	36
2.6	Conjunto linearmente separável.	41
2.7	Conjunto não linearmente separável.	44
2.8	Separação de um conjunto de dados usando variáveis de folga.	45
2.9	Vetor (w_0, b_0, ξ) satisfazendo (2.15) e (2.16).	46
2.10	Conjunto não linearmente separável.	62
2.11	Hiperplano e curva descrita pela função ϕ , no espaço de características.	63
2.12	Obtenção do hiperplano.	63
2.13	Classificação no espaço de entrada.	64
2.14	Similaridade entre os dados.	74
3.1	Hiperplano separador para um conjunto linearmente separável.	77
3.2	Hiperplano separador com conjunto de teste.	80
3.3	Dados linearmente separáveis, para aplicação do Algoritmo 2.1.	82
3.4	Eliminação de dados pelo Algoritmo 2.1.	83
3.5	Hiperplanos para somente dois dados, não perpendiculares.	83
3.6	Dados de teste representados pelos círculos abertos. Hiperplanos separadores obtidos com parâmetros $C = 10^{-3}$, $C = 10^{-1}$, $C = 1$ e $C = 10$, respectivamente.	85
3.7	Conjunto de dados não linearmente separável.	86
3.8	Região em azul: classe positiva. Região em vermelho: classe negativa. Superfície separadora no espaço de entrada para o mapeamento ϕ_1	87
3.9	Hipérbole $F(x) = 0$	88
3.10	Exemplo de uma imagem digital e intensidades de alguns dos seus pixels.	88

3.11 Criação dos caracteres e da imagem, respectivamente.	89
3.12 Conversão da imagem em matriz.	90
3.13 Imagem RGB.	91

Lista de Tabelas

2.1	Exemplos de funções <i>Kernel</i>	73
3.1	Matriz de confusão para 2 Classes.	80
3.2	Matriz de confusão para conjunto de teste.	81
3.3	Soluções obtidas pelo Algoritmo 2.1.	81
3.4	Soluções obtidas pelo Algoritmo 2.1.	82
3.5	Resultados para diferentes valores do parâmetro C	84
3.6	Acurácia para diferentes valores do parâmetro C	85
3.7	Resultados para mapeamentos distintos.	87
3.8	Matriz de variância covariância.	92

Lista de abreviaturas e siglas

VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	Falso Negativo
KKT	Karush-Kuhn-Tucker
SVM	<i>Support Vector Machine</i>
PCA	<i>Principal Component Analysis</i>
RBF	<i>Radial Basis Function</i>
RGB	<i>Red, Green and Blue</i>

Lista de símbolos

Os principais símbolos utilizados nesta dissertação são os seguintes.

X	Conjunto de dados de treinamento.
n	Dimensão do espaço de entrada.
N	Dimensão do espaço de características.
C	Parâmetro de ponderação dos erros.
x_i	i -ésima componente do vetor x .
y_i	Saídas.
m	Número de pontos de treinamento.
\mathbb{R}^n	Espaço euclidiano n -dimensional.
$ X $	Cardinalidade do conjunto X .
x^i	Ponto i do espaço de entrada.
ξ_i	Variável de folga do ponto x^i .
$\ w\ $	Norma euclidiana de $w \in \mathbb{R}^n$.
x^T	Transposto do vetor x .
$x^T y$	Produto interno euclidiano entre $x, y \in \mathbb{R}^n$.
$\frac{\partial f}{\partial w}$	Derivada parcial de f em relação à variável w .
α_i	Multiplicador de Lagrange associado à restrição de desigualdade.
β_i	Multiplicador de Lagrange associado à restrição de desigualdade.
e	Vetor cujas componentes são todas iguais a 1.
σ	Parâmetro do <i>Kernel</i> Gaussiano.
p	Parâmetro de homogeneidade do <i>Kernel</i> Polinomial.
d	Parâmetro de grau do <i>Kernel</i> Polinomial.
$K(x, y)$	Função <i>Kernel</i> entre os pontos x e y .

Outras notações serão introduzidas ao longo do texto quando se fizerem necessárias.

Sumário

Introdução	3
1 Revisão de Conceitos	7
1.1 Conceitos de Análise e Álgebra Linear	7
1.2 Conceitos de Otimização	12
1.2.1 Otimização convexa	14
1.2.2 Programação quadrática	18
1.2.3 Dualidade	20
2 Máquinas de Vetores Suporte	26
2.1 Metodologia de Aprendizagem	26
2.2 Máquinas de Vetores Suporte - Margem rígida	30
2.3 Máquinas de Vetores Suporte - CSVM	43
2.4 Máquinas de Vetores Suporte - Não linear	62
3 Experimentos numéricos	75
3.1 Metodologia	75
3.1.1 Implementação SVM - Margem rígida	76
3.1.2 Aplicação com Algencan	81
3.1.3 Implementação SVM - com margem flexível	84
3.1.4 Implementação SVM - não linear	85
3.2 Classificação de caracteres	88
3.2.1 Critérios de classificação	93
3.2.2 Escolha dos parâmetros	94
4 Conclusões e trabalhos futuros	97
Apêndice	98
A Códigos em Matlab	99
Referências Bibliográficas	110

Introdução

Vários campos da ciência fazem uso da Otimização para ajudar na tomada de decisões. Em particular, isso é observado na Aprendizagem de Máquina (do inglês *Machine Learning*), que tem apresentado a intenção de criar um algoritmo tão eficiente em decisão quanto o cérebro humano.

Nesta dissertação, apresentamos primeiramente uma introdução à Aprendizagem de Máquina, que é um dos campos mais procurados da inteligência computacional recentemente [24]. Ela está presente, por exemplo, quando fazemos uma pesquisa no Google ou para o reconhecimento de amigos no Facebook.

Existem dois tipos de Aprendizagem de Máquina, a supervisionada, que a partir de casos solucionados deduz o caso desconhecido, e a não supervisionada, em que a máquina cria sua própria solução [3, 13]. Neste trabalho, temos como objetivo realizar um estudo teórico e uma pequena aplicação de uma técnica supervisionada: as Máquinas de Vetores Suporte (SVM, do Inglês *Support Vector Machine*).

A técnica SVM foi desenvolvida por Vladimir Vapnik, Bernhard Boser, Isabelle Guyon e Corrina Cortes, com seus fundamentos provenientes da Teoria de Aprendizagem Estatística [21, 22, 34], que vem despertando interesse nos últimos anos [2, 9, 10, 15]. Conta hoje com vários trabalhos publicados, como por exemplo [1, 4, 5, 21, 28]. Os livros de Nello Cristianini e John Shawe-Taylor [15, 16] são referências muito utilizadas, na primeira tem uma introdução para tal técnica. Na sequência apresentamos as principais ideias sobre SVM.

Tal técnica concentra a atenção no seguinte problema de programação quadrática, convexa e com restrições

$$\begin{aligned} \min_{w,b} \quad & f(w, b) \quad \text{com } w \in \mathbb{R}^n \text{ e } b \in \mathbb{R} \\ \text{s.a.} \quad & g(w, b) \leq 0, \end{aligned} \tag{1}$$

onde as funções $f : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ e $g : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ são continuamente diferenciáveis. Neste trabalho, usamos SVM para classificação. O objetivo é obter um classificador linear $\text{sign}(w^T x + b)$, a partir da solução (w, b) . A Figura 1(a) representa um problema de classificação que pode ser modelado pelo problema (1), o qual estamos interessados em resolver. A partir da solução deste problema construímos um hiperplano definido pela

equação $w^T x + b = 0$, que será usado para classificação de novos dados.

Tratamos com detalhes da modelagem SVM com margem rígida [34] e suas generalizações, que são margem flexível [21] e não linear [22], conforme Figura 1.

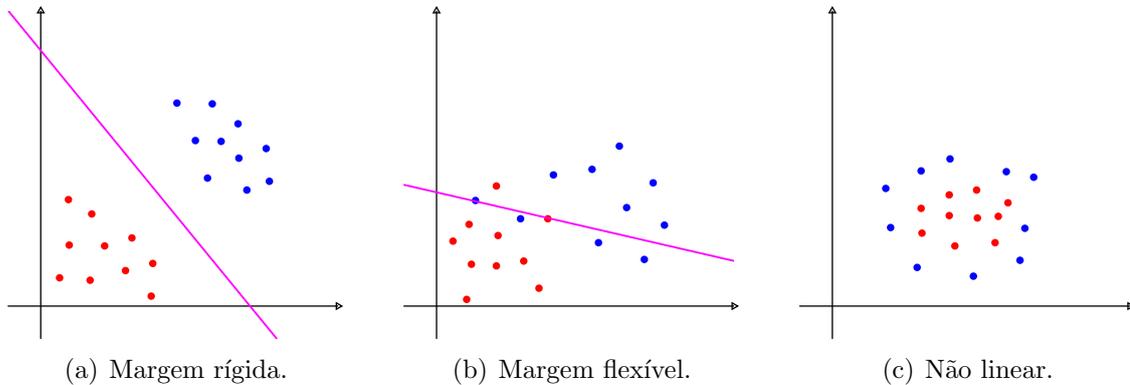


Figura 1: Conjunto de dados para classificação.

Para fundamentar o desenvolvimento de SVM, destacamos no Capítulo 1 alguns resultados clássicos de Análise, Álgebra Linear e Otimização. Apresentamos a definição de espaço de Hilbert, já que precisamos deste conceito para tratarmos do chamado Truque do *Kernel*, usado em SVM não linear. Como a técnica envolve um problema convexo, os pontos estacionários são soluções globais. Entretanto, só a convexidade não garante a existência de um minimizador, nem mesmo com a hipótese de limitação da função objetivo, conforme exemplos apresentados no texto. Para um problema quadrático, tratamos de condições que garantem a existência de solução global, baseados no trabalho de Blum e Oettli [23]. A prova deste resultado foi detalhada e apresentada com algumas modificações. Também discutimos a questão da unicidade da solução para problemas convexos.

Este trabalho tem âmbito teórico, analisamos pesquisas publicadas sobre SVM, buscando prover conceitos e teorias essenciais desta técnica, procurando apresentar da forma mais didática possível. Uma das principais contribuições consiste na fundamentação teórica da existência e unicidade para obtenção do hiperplano ótimo.

A principal característica de SVM é a utilização da teoria de dualidade Lagrangiana [17, 19], que facilita os cálculos para obter o classificador ótimo, já que utiliza somente alguns dados, conhecidos como vetores suporte, daí o nome da técnica. Reescrevemos resultados que usam o *gap* de dualidade sem precisar do conceito de ponto de sela.

Para o dual, podemos garantir a existência de solução, para a SVM com margem rígida. Tratamos sobre a obtenção de vetores suporte, pois não podemos garantir que o problema dual tem solução única e mostramos um exemplo para o qual o dual possui infinitas soluções. À luz deste exemplo, apresentamos duas definições de vetores suporte bastante presentes na literatura [1, 3, 15, 35]. Provamos que pontos associados a restrições

inativas são desnecessários para a obtenção da solução primal. Mostramos que o *gap* de dualidade é zero, o que permite estabelecer uma relação entre uma solução qualquer do dual e a solução do primal. Por conta desta relação, podemos simplificar, tanto teoricamente quanto computacionalmente, a obtenção do hiperplano ótimo e assim propomos um algoritmo que gera uma quantidade consideravelmente pequena de vetores suporte. Para sua utilização, recorreremos à interface do Algencan em Matlab [29, 31].

Novamente, para o problema SVM com margem flexível, podemos garantir a existência de solução do problema primal. Diferentemente do SVM com margem rígida, somente sob certas condições podemos garantir a unicidade da solução. No entanto, quando existem mais soluções, todas elas definem hiperplanos paralelos. Reestruturamos os enunciados e as demonstrações a respeito da existência e unicidade da solução, tanto do problema primal quanto do dual, baseados em [25].

Para SVM não linear, todos os resultados estabelecidos anteriormente, em particular os que tratam da existência e unicidade, se aplicam para o problema primal, bastando trocar os dados pelo mapeamento realizado. Tal mapeamento tem como objetivo obter um hiperplano ótimo em um espaço de dimensão elevada. Entretanto, recorreremos ao problema dual, para usarmos as funções *Kernel*. Sentimos falta na literatura [3, 15, 20] da discussão sobre o cálculo de b quando não temos unicidade da solução primal.

Apresentamos duas definições de funções *kernel*, uma delas usando mapeamento para um espaço de dimensão finita. A outra, mais geral, considerando mapeamentos em espaços de Hilbert. Mostramos que estas duas definições não são equivalentes. Discutimos diversas propriedades, bem como formas de caracterizar uma função *Kernel*, em particular o Teorema de Mercer.

As aplicações de SVM são para inúmeros problemas práticos, tais como reconhecimento facial [32], leitura de placas automotivas [30], previsões de bolsa [33], detecção de spam [8], detecção de faces humanas em imagens [7], etc. O objetivo específico deste trabalho é realizar uma pequena aplicação sobre o problema de classificação de caracteres. Aplicamos a técnica, visando analisar os resultados com a teoria apresentada. Levamos em conta a escolha dos parâmetros da função *kernel* Gaussiana e do parâmetro de regularização C , abordado no caso flexível.

Principais contribuições

Em resumo, as principais contribuições da dissertação são:

- Apresentamos o Teorema 1.31, que garante a unicidade de um problema convexo.
- Deixamos acessível o resultado que não é tão clássico: Teorema 1.34, que trata da existência de solução global para programação quadrática.

- Reformulamos os resultados de dualidade, que usualmente recorrem ao conceito de ponto de sela.
- Fundamentamos teoricamente a existência e a unicidade das soluções dos problemas primal e dual, obtidos em SVM.
- Discutimos a respeito das definições de vetores suporte, presentes na literatura [1, 3, 15, 35], tentando deixar clara a diferença entre elas.
- Provamos que ao retirar pontos que não são vetores suporte o classificador não muda.
- Propomos o Algoritmo 2.1, que busca uma solução dual o mais esparsa possível.
- Apresentamos modificações nos resultados sobre a unicidade da solução primal, para SVM com margem flexível, encontrados em [25].
- Discutimos duas definições (não equivalentes) de funções *Kernel* e enunciamos e demonstramos diversos resultados sobre o tema.
- Implementamos o Algoritmo 2.1, recorrendo à interface do Algencan em Matlab.
- Realizamos testes numéricos para o problema de classificação de caracteres, acompanhados da análise na escolha do parâmetro de regularização C , da função *Kernel* e seus respectivos parâmetros.

Organização do trabalho

Apresentamos o estudo da seguinte forma. No Capítulo 1, onde as principais referências foram [14, 17, 19, 20, 26], apresentamos algumas definições e teoremas, que são utilizadas para fundamentar o desenvolvimento da técnica SVM.

No Capítulo 2, apresentamos na Seção 2.1 uma revisão de literatura, com base em [3, 4, 5, 15, 28]. Depois, explicamos com detalhes, na Seção 2.2, a modelagem da técnica SVM linear, na Seção 2.3 discutimos SVM com margem flexível e na Seção 2.4, SVM com o chamado Truque do *Kernel*, baseados sempre na teoria de dualidade Lagrangiana.

No Capítulo 3, realizamos experimentos numéricos mostrando os detalhes da implementação. Além disso, uma vez que tratamos SVM para problemas de classificação binária, mostramos como utilizá-la em problemas multinomiais, recorrendo às estratégias *one-against-one* e *one-against-rest* [3].

Finalmente, no Capítulo 4, temos as principais conclusões de todo o estudo realizado e as sugestões para trabalhos futuros. Os códigos em Matlab, elaborados para resolver os problemas apresentados no texto recorrendo à interface do Algencan [29, 31] e algumas funções próprias do Matlab, compõem o Apêndice A.

Capítulo 1

Revisão de Conceitos

Neste capítulo destacamos algumas definições clássicas e alguns resultados de Análise, Álgebra Linear e Otimização para melhor entendimento deste trabalho. Também apresentamos algumas definições e teoremas necessários para fundamentar o desenvolvimento da técnica SVM, que requer a solução de um problema de Otimização. As principais referências consideradas neste capítulo são [14, 17, 19, 20, 26].

1.1 Conceitos de Análise e Álgebra Linear

Lema 1.1 [14, pág. 50] *Considere $\|\cdot\|$ a norma euclidiana em \mathbb{R}^n . Dados $a, b \in \mathbb{R}^n$ com $a \neq b$ tais que $\|a\| = \|b\| = r$, então $\|(1-t)a + tb\| < r$ para todo $t \in (0, 1)$.*

Teorema 1.2 *Toda sequência convergente é limitada.*

Teorema 1.3 [Bolzano-Weierstrass] *Toda sequência limitada em \mathbb{R}^n possui uma subsequência convergente.*

Definição 1.4 *Seja $A \in \mathbb{R}^{n \times n}$ uma matriz simétrica. Dizemos que A é definida positiva quando $x^T A x > 0$, para todo $x \in \mathbb{R}^n \setminus \{0\}$. Se $x^T A x \geq 0$, para todo $x \in \mathbb{R}^n$, A é dita semidefinida positiva.*

Proposição 1.5 *Uma matriz $A \in \mathbb{R}^{n \times n}$ é semidefinida positiva se, e somente se, para alguma matriz real B , temos que $A = B^T B$.*

Teorema 1.6 [Teorema Espectral] *Seja $A \in \mathbb{R}^{n \times n}$ uma matriz simétrica. Então*

1. *os autovalores $\lambda_1, \dots, \lambda_n$ de A são reais.*
2. *os autovetores de A associados a autovalores distintos são ortogonais.*
3. *é sempre possível obter uma base ortonormal de autovetores $\{v_1, \dots, v_n\}$.*
4. *vale a relação $VDV^T = A$, com D matriz diagonal com autovalores de A , as colunas de V são v_1, \dots, v_n e, portanto, V é uma matriz ortogonal.*

Proposição 1.7 *Sejam $A, B \in \mathbb{R}^{n \times n}$ simétricas e semidefinidas positivas. Definimos o produto de Hadamard de A por B como*

$$A \circ B = (c_{ij}) \in \mathbb{R}^{n \times n},$$

onde $c_{ij} = a_{ij}b_{ij}$. Então $A \circ B$ é simétrica e semidefinida positiva.

Demonstração. Considere as decomposições espectrais $A = UD_1U^T$ e $B = VD_2V^T$, com $U = (u^1, \dots, u^n)$, $D_1 = \text{diag}(\lambda_1, \dots, \lambda_n)$, $V = (v^1, \dots, v^n)$ e $D_2 = \text{diag}(\mu_1, \dots, \mu_n)$. Isto significa que

$$A = \sum_{k=1}^n \lambda_k u^k (u^k)^T \text{ e } B = \sum_{l=1}^n \mu_l v^l (v^l)^T.$$

Afirmamos que

$$A \circ B = \sum_{k,l=1}^n \lambda_k \mu_l (u^k \circ v^l) (u^k \circ v^l)^T, \quad (1.1)$$

onde $u \circ v$ também indica o vetor produto termo a termo. Para provar esta afirmação, considere um elemento arbitrário $c_{ij} = a_{ij}b_{ij}$ de $A \circ B$. Como

$$a_{ij} = e_i^T A e_j = \sum_{k=1}^n \lambda_k u_i^k u_j^k \text{ e } b_{ij} = e_i^T B e_j = \sum_{l=1}^n \mu_l v_i^l v_j^l,$$

temos

$$c_{ij} = \sum_{k,l=1}^n \lambda_k \mu_l u_i^k u_j^k v_i^l v_j^l = \sum_{k,l=1}^n \lambda_k \mu_l (u^k \circ v^l)_i (u^k \circ v^l)_j,$$

o que prova (1.1).

Agora, para ver que $A \circ B$ é semidefinida positiva, note que

$$x^T (A \circ B) x = \sum_{k,l=1}^n \lambda_k \mu_l x^T (u^k \circ v^l) (u^k \circ v^l)^T x = \sum_{k,l=1}^n \lambda_k \mu_l (x^T (u^k \circ v^l))^2 \geq 0.$$

□

Proposição 1.8 [*Desigualdade de Cauchy-Schwarz*] *Seja $(H, \langle \cdot, \cdot \rangle)$ um espaço com produto interno e $\| \cdot \|$ a norma por ele induzida. Então, para quaisquer $x, y \in H$, temos*

$$| \langle x, y \rangle | \leq \|x\| \|y\|,$$

onde a igualdade ocorre se, e somente se, os vetores x e y são linearmente dependentes.

Uma sequência $(x^k)_{k \in \mathbb{N}}$ em um espaço métrico é dita uma sequência de Cauchy quando para todo $\epsilon > 0$ existe $n_0 \in \mathbb{N}$ tal que se $k, j \geq n_0$ então $\|x^k - x^j\| < \epsilon$.

Definição 1.9 *Um espaço métrico é completo quando toda sequência de Cauchy converge para um limite que pertence ao espaço.*

Exemplo 1.10 Um exemplo de espaço completo é o \mathbb{R}^n .

Definição 1.11 Um conjunto \mathcal{H} é um Espaço de Hilbert se satisfaz as seguintes condições:

1. \mathcal{H} é um espaço vetorial sobre \mathbb{C} ou \mathbb{R} .
2. \mathcal{H} é munido de um produto interno $\langle \cdot, \cdot \rangle$.
3. \mathcal{H} é completo em relação à métrica induzida por esse produto interno.
4. \mathcal{H} é separável.

Exemplo 1.12 O conjunto das seqüências com quadrado somável

$$\ell_2 = \left\{ (\psi_k)_{k \in \mathbb{N}} \mid \sum_{k=1}^{\infty} \psi_k^2 < \infty \right\}$$

com o produto interno

$$\langle \psi, \zeta \rangle = \sum_{k=1}^{\infty} \psi_k \zeta_k$$

é um espaço de Hilbert. Mostraremos aqui que a propriedade 2 da Definição 1.11 vale.

De fato, se $s_1 = \sum_{k=1}^{\infty} \psi_k^2 < \infty$ e $s_2 = \sum_{k=1}^{\infty} \zeta_k^2 < \infty$, então como $|\psi_k \zeta_k| \leq \frac{1}{2}(\psi_k^2 + \zeta_k^2)$, temos

$$\sum_{k=1}^r |\psi_k \zeta_k| = \sum_{k=1}^r \frac{1}{2}(\psi_k^2 + \zeta_k^2) \leq \frac{1}{2}(s_1 + s_2) < \infty$$

para todo $r \in \mathbb{N}$, o que prova que a série $\sum_{k=1}^{\infty} \psi_k \zeta_k$ é (absolutamente) convergente.

O próximo resultado será importante para estabelecer uma propriedade muito útil das funções *Kernel*, que serão estudadas no Capítulo 2. Para tanto, considere um conjunto compacto $\mathcal{X} \subset \mathbb{R}^n$ e o espaço

$$L_2(\mathcal{X}) = \left\{ f : \mathcal{X} \rightarrow \mathbb{R} \mid \int_{\mathcal{X}} (f(x))^2 dx < \infty \right\}.$$

Teorema 1.13 Sejam $\mathcal{X} \subset \mathbb{R}^n$ um conjunto compacto e $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ uma função contínua e simétrica. Dados $x^1, \dots, x^m \in \mathcal{X}$, $c_1, \dots, c_m \in \mathbb{R}$ e $\varepsilon > 0$, existe $f \in L_2(\mathcal{X})$ tal que

$$\left| \int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz - \sum_{i,j=1}^m K(x^i, x^j) c_i c_j \right| \leq \varepsilon.$$

Demonstração. Vamos provar o teorema no caso em que \mathcal{X} é um bloco em \mathbb{R}^n , isto é, um produto cartesiano de intervalos compactos de \mathbb{R} . Como K é contínua, podemos

considerar blocos $B_i \subset \mathcal{X}$, $i = 1, \dots, m$, tais que $x^i \in B_i$, $B_i \cap B_j = \emptyset$ e

$$\sup_{(x,z) \in B_i \times B_j} \{|(K(x,z) - K(x^i, x^j))c_i c_j|\} \leq \frac{\varepsilon}{m^2}$$

para todos $i, j = 1, \dots, m$. Defina então $f : \mathcal{X} \rightarrow \mathbb{R}$ como

$$f(x) = \begin{cases} \frac{c_i}{\text{vol}(B_i)}, & \text{se } x \in B_i, \\ 0, & \text{c.c.} \end{cases}$$

Desta forma, temos que $f \in L_2(\mathcal{X})$ e

$$\int_{\mathcal{X} \times \mathcal{X}} K(x,z)f(x)f(z)dx dz = \sum_{i,j=1}^m \int_{B_i \times B_j} K(x,z)f(x)f(z)dx dz. \quad (1.2)$$

Além disso, fixados $i, j = 1, \dots, m$, temos

$$\left| K(x,z)f(x)f(z) - \frac{K(x^i, x^j)c_i c_j}{\text{vol}(B_i)\text{vol}(B_j)} \right| = \frac{|(K(x,z) - K(x^i, x^j))c_i c_j|}{\text{vol}(B_i)\text{vol}(B_j)} \leq \frac{\varepsilon}{m^2 \text{vol}(B_i)\text{vol}(B_j)}$$

para todo $(x,z) \in B_i \times B_j$. Portanto, denotando $I_{ij} = \int_{B_i \times B_j} K(x,z)f(x)f(z)dx dz$ e $\gamma_{ij} = K(x^i, x^j)c_i c_j$, temos

$$\begin{aligned} |I_{ij} - \gamma_{ij}| &= \left| \int_{B_i \times B_j} \left(K(x,z)f(x)f(z) - \frac{K(x^i, x^j)c_i c_j}{\text{vol}(B_i \times B_j)} \right) dx dz \right| \\ &\leq \int_{B_i \times B_j} \left| \left(K(x,z)f(x)f(z) - \frac{K(x^i, x^j)c_i c_j}{\text{vol}(B_i \times B_j)} \right) \right| dx dz \\ &\leq \frac{\varepsilon}{m^2 \text{vol}(B_i)\text{vol}(B_j)} \text{vol}(B_i \times B_j) \\ &= \frac{\varepsilon}{m^2}. \end{aligned}$$

Assim, em vista de (1.2) temos

$$\begin{aligned} \left| \int_{\mathcal{X} \times \mathcal{X}} K(x,z)f(x)f(z)dx dz - \sum_{i,j=1}^m K(x^i, x^j)c_i c_j \right| &= \left| \sum_{i,j=1}^m (I_{ij} - \gamma_{ij}) \right| \\ &\leq \sum_{i,j=1}^m |I_{ij} - \gamma_{ij}| \\ &= \varepsilon, \end{aligned}$$

provando o teorema. □

Corolário 1.14 *Sejam $\mathcal{X} \subset \mathbb{R}^n$ um conjunto compacto e $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ uma função*

contínua e simétrica tal que

$$\int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz \geq 0$$

para toda $f \in L_2(\mathcal{X})$. Então, dados $x^1, \dots, x^m \in \mathcal{X}$ e $c_1, \dots, c_m \in \mathbb{R}$, temos

$$\sum_{i,j=1}^m K(x^i, x^j) c_i c_j \geq 0.$$

Vejamos que a recíproca do corolário também é verdadeira. Mais precisamente, temos o seguinte resultado.

Teorema 1.15 *Sejam $\mathcal{X} \subset \mathbb{R}^n$ um conjunto compacto e $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ uma função contínua e simétrica. Suponha que*

$$\sum_{i,j=1}^m K(x^i, x^j) c_i c_j \geq 0$$

para todos $x^1, \dots, x^m \in \mathcal{X}$, $m \in \mathbb{N}$ e $c_1, \dots, c_m \in \mathbb{R}$. Então,

$$\int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz \geq 0$$

para toda $f \in L_2(\mathcal{X})$.

Demonstração. Suponha por absurdo que para alguma função $f \in L_2(\mathcal{X})$ e $\varepsilon > 0$ tenhamos

$$\int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz < -\varepsilon.$$

Pelas propriedades de integrais múltiplas, existe $\delta > 0$ tal que para qualquer decomposição $\mathcal{X} \times \mathcal{X} = \bigcup_{\ell \in \mathcal{L}} A_\ell$ satisfazendo $\sup\{\|p - q\| \mid p, q \in A_\ell, \ell \in \mathcal{L}\} < \delta$ e quaisquer pontos $(x^\ell, z^\ell) \in A_\ell$, temos

$$\left| \int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz - \sum_{\ell \in \mathcal{L}} K(x^\ell, z^\ell) f(x^\ell) f(z^\ell) \text{vol}(A_\ell) \right| < \varepsilon.$$

Seja uma decomposição $\mathcal{X} = \bigcup_{i=1}^m B_i$ tal que

$$\sup\{\|x - u\| \mid x, u \in B_i, i = 1, \dots, m\} < \frac{\delta}{\sqrt{2}}.$$

Temos que $\|(x, z) - (u, v)\| < \delta$ para todos $(x, z), (u, v) \in B_i \times B_j$ e $i, j = 1, \dots, m$.

Portanto, para a decomposição $\mathcal{X} \times \mathcal{X} = \bigcup_{i,j=1}^m B_i \times B_j$ e pontos $x^i \in B_i$ temos

$$\left| \int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz - \sum_{i,j=1}^m K(x^i, x^j) f(x^i) f(x^j) \text{vol}(B_i \times B_j) \right| < \varepsilon.$$

Como $\text{vol}(B_i \times B_j) = \text{vol}(B_i) \text{vol}(B_j)$, definindo $c_i = f(x^i) \text{vol}(B_i)$, chegamos em

$$\sum_{i,j=1}^m K(x^i, x^j) c_i c_j < 0,$$

contradizendo a hipótese. □

1.2 Conceitos de Otimização

Vamos discutir nesta seção alguns conceitos e resultados de Otimização para o problema

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.a.} \quad & g(x) \leq 0, \\ & h(x) = 0, \end{aligned} \tag{1.3}$$

onde a função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e as funções que definem as restrições $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ são continuamente diferenciáveis. No que segue, denotamos o conjunto viável do problema (1.3) por

$$\Omega \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid g(x) \leq 0, h(x) = 0\}.$$

Definição 1.16 Dizemos que x^* é uma solução global do problema (1.3), ou seja, um minimizador global de f em Ω , quando $f(x^*) \leq f(x)$ para todo $x \in \Omega$.

Dado $\bar{x} \in \Omega$, dizemos que uma restrição de desigualdade g_i é ativa em \bar{x} quando $g_i(\bar{x}) = 0$. Caso $g_i(\bar{x}) < 0$, dizemos que g_i é inativa em \bar{x} . O conjunto dos índices das restrições de desigualdade ativas é denotado por

$$I(\bar{x}) = \{i \mid g_i(\bar{x}) = 0\}.$$

Definição 1.17 Um ponto $x^* \in \mathbb{R}^n$ é dito estacionário para o problema (1.3) quando

existirem vetores $\alpha^* \in \mathbb{R}^m$, $\beta^* \in \mathbb{R}^p$ (multiplicadores de Lagrange) tais que

$$\nabla f(x^*) + \sum_{i=1}^m \alpha_i^* \nabla g_i(x^*) + \sum_{i=1}^p \beta_i^* \nabla h_i(x^*) = 0, \quad (1.4a)$$

$$g(x^*) \leq 0, h(x^*) = 0, \quad (1.4b)$$

$$\alpha_i^* \geq 0, \quad i = 1, \dots, m, \quad (1.4c)$$

$$\alpha_i^* g_i(x^*) = 0, \quad i = 1, \dots, m. \quad (1.4d)$$

As condições (1.4a)–(1.4d) são conhecidas como condições de Karush-Kuhn-Tucker (KKT) e, sob determinadas hipóteses de qualificação, são satisfeitas em um ponto que seja minimizador.

Um ponto $x \in \mathbb{R}^n$ é dito qualificado quando atende uma condição de qualificação. Apresentamos a seguir algumas dessas condições de qualificação para o problema (1.3).

Condição de qualificação de independência linear: Dizemos que a condição de qualificação de independência linear (LICQ) é satisfeita em \bar{x} quando o conjunto formado pelos gradientes das restrições de igualdade e das restrições de desigualdade ativas é linearmente independente, isto é,

$$\{\nabla g_i(\bar{x}) \mid i \in I(\bar{x})\} \cup \{\nabla h_i(\bar{x}), i = 1, \dots, p\} \text{ é LI.}$$

Condição de qualificação de Slater: Dizemos que a condição de qualificação de Slater é satisfeita quando h é afim, cada componente g_i , $i = 1, \dots, m$, é convexa e existe $\tilde{x} \in \Omega$ tal que $h(\tilde{x}) = 0$ e $g(\tilde{x}) < 0$.

Teorema 1.18 (KKT) *Seja $x^* \in \mathbb{R}^n$ um minimizador local do problema (1.3) e suponha que seja satisfeita uma condição de qualificação. Então existem vetores $\alpha^* \in \mathbb{R}^m$, $\beta^* \in \mathbb{R}^p$ tais que (x^*, α^*, β^*) cumpre (1.4a)–(1.4d).*

Note que se não for verificada nenhuma condição de qualificação, podemos ter minimizadores que não cumpram KKT, atrapalhando a caracterização de tais pontos.

Existe uma situação particular, que será muito importante para o Capítulo 2, em que as restrições são afins, caso em que os pontos viáveis também são qualificados.

Conjunto viável poliedral: Suponha que as restrições g e h são afins, isto é, digamos que o problema seja da forma

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.a.} \quad & Ax \leq b \\ & Mx = r, \end{aligned}$$

onde $A \in \mathbb{R}^{m \times n}$, $M \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^m$ e $r \in \mathbb{R}^p$.

Neste caso, todo ponto viável é qualificado. Portanto, se x^* é um minimizador local deste problema, então x^* satisfaz as condições de KKT.

1.2.1 Otimização convexa

Uma hipótese com ótimas consequências em Otimização é a convexidade, que garante que pontos estacionários são soluções globais. Conforme veremos nesta dissertação, os problemas de classificação podem ser formulados em termos de Otimização convexa. Inicialmente definimos os conceitos básicos de convexidade.

Definição 1.19 [14, pág. 49] *Um conjunto $C \subset \mathbb{R}^n$ é dito convexo quando dados $x, y \in C$, o segmento $[x, y] = \{(1-t)x + ty \mid t \in [0, 1]\}$ estiver inteiramente contido em C .*

A noção de convexidade de conjuntos pode ser representada pela Figura 1.1, que ilustra dois conjuntos, um convexo e outro não. Notamos que, no segundo conjunto nem todo segmento que liga dois pontos do conjunto está inteiramente contido no conjunto.

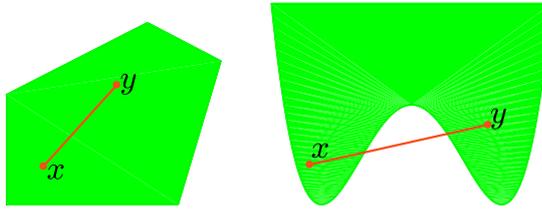


Figura 1.1: Exemplo de conjunto convexo e não convexo.

Definição 1.20 [14, pág. 56] *Seja $C \subset \mathbb{R}^n$ um conjunto convexo. Dizemos que a função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é convexa em C quando*

$$f((1-t)x + ty) \leq (1-t)f(x) + tf(y),$$

para todos $x, y \in C$ e $t \in [0, 1]$. Caso a desigualdade seja estrita, para todos $x, y \in C$ e $t \in (0, 1)$, f é dita estritamente convexa.

A Figura 1.2 mostra a noção geométrica da definição citada acima, em que qualquer arco no gráfico de uma função convexa está sempre abaixo do segmento que liga as extremidades.

Observação 1.21 *Todo problema de maximização*

$$\begin{aligned} \max_x & f(x) \\ \text{s.a} & x \in \Omega \end{aligned}$$

pode ser transformado em um problema de minimização equivalente

$$\begin{aligned} \min_x & -f(x) \\ \text{s.a} & x \in \Omega. \end{aligned}$$

Observação 1.22 *Uma função f é dita côncava se $-f$ for convexa. Assim, a maximização de uma função côncava equivale à minimização de uma função convexa.*

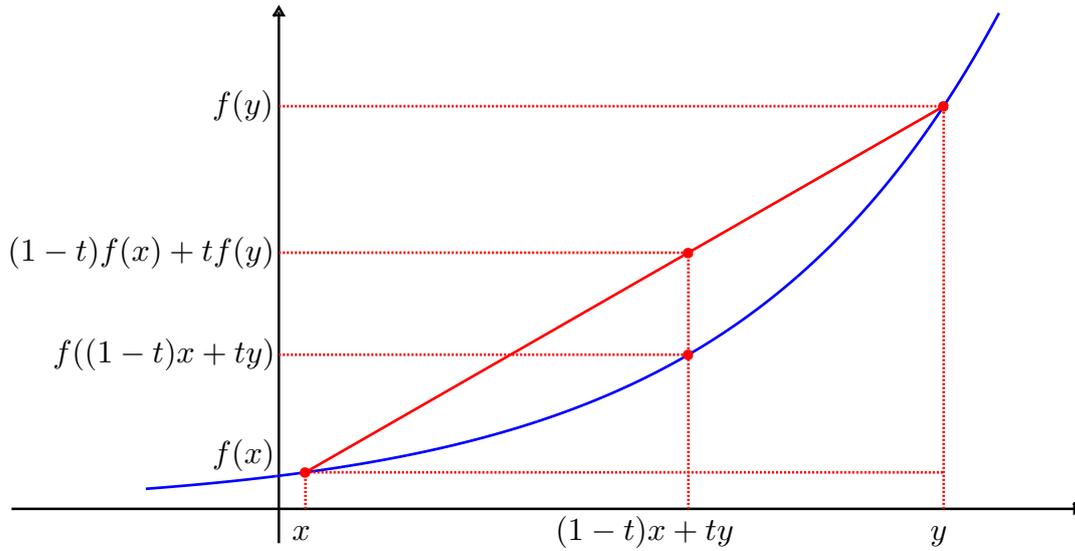


Figura 1.2: Função convexa.

O próximo resultado justifica o fato da convexidade ser uma propriedade tão interessante.

Teorema 1.23 *Considere o problema (1.3) e suponha que $\Omega \subset \mathbb{R}^n$ é convexo e f é uma função convexa em Ω . Então todo minimizador local para este problema será minimizador global.*

Demonstração. Seja $x^* \in \Omega$ um minimizador local para o problema (1.3). Então existe $\delta > 0$ tal que $f(x^*) \leq f(x)$ para todo $x \in B(x^*, \delta) \cap \Omega$. Dado $y \in \Omega$, $y \notin B(x^*, \delta)$, tome $t \in (0, 1]$ de modo que $t\|y - x^*\| < \delta$. Assim, o ponto $x = (1 - t)x^* + ty$ satisfaz

$$\|x - x^*\| = t\|y - x^*\| < \delta$$

e portanto $x \in B(x^*, \delta) \cap \Omega$. Deste modo temos

$$f(x^*) \leq f(x) \leq (1 - t)f(x^*) + tf(y) = f(x^*) + t(f(y) - f(x^*)),$$

donde segue que $f(x^*) \leq f(y)$. □

Apresentamos a seguir outras formas de caracterizar a convexidade de uma função, quando temos hipóteses de diferenciabilidade.

Teorema 1.24 [14, pág. 58] *Sejam $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função diferenciável e $C \subset \mathbb{R}^n$ um conjunto convexo. A função f é convexa em C se, e somente se,*

$$f(y) \geq f(x) + \nabla f(x)^T(y - x)$$

para todos $x, y \in C$.

Teorema 1.25 [14, pág. 60] *Sejam $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função de classe \mathcal{C}^2 e $C \subset \mathbb{R}^n$ um conjunto convexo com interior não vazio. A função f é convexa em C se, e somente se, a Hessiana $\nabla^2 f(x)$ é semidefinida positiva para todo $x \in C$.*

O próximo resultado estabelece que pontos estacionários de problemas convexos são minimizadores globais.

Teorema 1.26 *Considere o problema (1.3) e suponha que $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, são funções continuamente diferenciáveis e convexas. Suponha também que h é afim. Então, qualquer ponto estacionário para este problema é minimizador global.*

Demonstração. Considere x^* um ponto estacionário. Usando a convexidade de g_i , obtemos

$$\nabla g_i(x^*)^T(x - x^*) = g_i(x^*) + \nabla g_i(x^*)^T(x - x^*) \leq g_i(x) \leq 0,$$

para todo $i \in I(x^*)$ e $x \in \Omega$. Além disso, como h é afim, temos $h_i(x) = v_i^T x + b$, donde segue que

$$\nabla h_i(x^*)^T(x - x^*) = v_i^T(x - x^*) = (v_i^T x + b) - (v_i^T x^* + b) = 0,$$

para todo $i = 1, \dots, p$ e $x \in \Omega$. Portanto, em vista de (1.4a), (1.4c) e (1.4d), obtemos

$$\nabla f(x^*)^T(x - x^*) = - \sum_{i \in I(x^*)} \alpha_i^* \nabla g_i(x^*)^T(x - x^*) - \sum_{i=1}^p \beta_i^* \nabla h_i(x^*)^T(x - x^*) \geq 0.$$

Assim, pela convexidade de f , temos que

$$f(x^*) \leq f(x^*) + \nabla f(x^*)^T(x - x^*) \leq f(x),$$

para todo $x \in \Omega$. □

Portanto, se o problema de Otimização é convexo, isto é, a função objetivo é convexa e as restrições determinam um conjunto convexo, então as condições de KKT são necessárias e suficientes para x^* ser minimizador global.

Salientamos que apenas a convexidade do problema não garante a existência de minimizador. Nem mesmo com a hipótese de limitação da função objetivo. Os exemplos a seguir justificam esta afirmação.

Exemplo 1.27 *Considere o problema convexo*

$$\begin{aligned} \min_x \quad & x_1^2 + x_2 \\ \text{s.a.} \quad & x_1 + x_2 - 1 \leq 0. \end{aligned}$$

Este problema não tem minimizador pois a função objetivo é ilimitada inferiormente. De fato, basta considerar $(x^k) = \begin{pmatrix} 0 \\ -k \end{pmatrix}$.

Exemplo 1.28 *O problema*

$$\begin{array}{ll} \min_x & e^{x_1+x_2} \\ \text{s.a} & x_1 + x_2 - 1 \leq 0 \end{array}$$

é convexo e a função objetivo é limitada inferiormente. No entanto, também não existe minimizador aqui.

Outro fato interessante em Otimização convexa é que o conjunto solução de um problema convexo é convexo.

Teorema 1.29 *Considere o problema (1.3) e suponha que $\Omega \subset \mathbb{R}^n$ é convexo e f é uma função convexa em Ω . Então o conjunto solução é convexo.*

Demonstração. Sejam $x^*, \tilde{x} \in \Omega$ duas soluções do problema (1.3). Então, em virtude do Teorema 1.23,

$$f(x^*) = f(\tilde{x}) = \gamma \stackrel{\text{def}}{=} \inf\{f(x) \mid x \in \Omega\}.$$

Portanto, para todo $t \in [0, 1]$,

$$f((1-t)x^* + t\tilde{x}) \leq (1-t)f(x^*) + tf(\tilde{x}) = \gamma,$$

provando que $(1-t)x^* + t\tilde{x}$ também é uma solução. □

A seguir, apresentaremos alguns resultados para garantir a unicidade de um problema convexo.

Teorema 1.30 *A solução para um problema convexo, para o qual a função objetivo é estritamente convexa, é única.*

Teorema 1.31 *Considere $\xi : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função estritamente convexa, $f : \mathbb{R}^{n+p} \rightarrow \mathbb{R}$ tal que $f(x, y) = \xi(x)$ e o seguinte problema de Otimização*

$$\begin{array}{ll} \min_{x,y} & f(x, y) \\ \text{s.a} & (x, y) \in \Omega, \end{array} \tag{1.5}$$

onde $\Omega \subset \mathbb{R}^{n+p}$ é um conjunto convexo. Seja $\Omega^* \subset \Omega$ o conjunto solução do problema (1.5) e suponha que existe uma função $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^p$ tal que $y = \phi(x)$ para todo $(x, y) \in \Omega^*$. Então o problema (1.5) possui uma única solução.

Demonstração. Considere $(x^*, y^*), (\tilde{x}, \tilde{y}) \in \Omega^*$. Então

$$\xi(x^*) = f(x^*, y^*) = f(\tilde{x}, \tilde{y}) = \xi(\tilde{x}).$$

Afirmamos que $x^* = \tilde{x}$. De fato, suponha que $x^* \neq \tilde{x}$. Como Ω^* é um conjunto convexo, temos que $(\hat{x}, \hat{y}) = \frac{1}{2}((x^*, y^*) + (\tilde{x}, \tilde{y})) \in \Omega^*$. Pela convexidade estrita de ξ , segue que

$\xi(\hat{x}) < \frac{1}{2}(\xi(x^*) + \xi(\tilde{x})) = \xi(x^*)$ e portanto, $f(\hat{x}, \hat{y}) < f(x^*, y^*)$, contradizendo o fato de (x^*, y^*) ser uma solução. Isto prova que $x^* = \tilde{x}$. Finalmente, note que

$$y^* = \phi(x^*) = \phi(\tilde{x}) = \tilde{y},$$

completando a prova. \square

Na próxima seção, trataremos de condições que garantem a existência de minimizador global para um problema quadrático.

1.2.2 Programação quadrática

Consideremos um problema de Otimização no qual a função objetivo é quadrática e o conjunto viável é um poliedro. Isto é,

$$\begin{aligned} \min_x \quad & f(x) = x^T H x + c^T x \\ \text{s.a} \quad & Ax + b \leq 0, \end{aligned} \tag{1.6}$$

onde $H \in \mathbb{R}^{n \times n}$ é simétrica, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$. Conforme Exemplo 1.27, este problema pode não ter solução. No entanto, se f for limitada inferiormente, então é garantida a existência de um minimizador. Este resultado não é muito clássico e se deve a Frank e Wolfe [18]. Por conveniência, será apresentada uma prova deste resultado devida a Blum e Oettli [23]. Na verdade, detalhamos a demonstração e realizamos algumas modificações visando corrigir alguns pontos.

Primeiramente, serão mostrados alguns resultados básicos a respeito de um problema geral de Otimização.

Lema 1.32 *O conjunto dos minimizadores globais do problema (1.3),*

$$S^* = \{x \in \Omega \mid f(x) \leq f(y), \forall y \in \Omega\},$$

é fechado.

Demonstração. Como o conjunto vazio é fechado, basta considerar o caso em que $S^* \neq \emptyset$. Suponha então que $(x^k) \subset S^*$ é tal que $x^k \rightarrow x^*$. Como $S^* \subset \Omega$, que é fechado, temos que $x^* \in \Omega$. Além disso, $f(x^k) \leq f(y)$ para todo $k \in \mathbb{N}$ e para todo $y \in \Omega$. Passando o limite obtemos $f(x^*) \leq f(y)$. \square

Lema 1.33 *Suponha que o conjunto S^* , definido no Lema 1.32, seja não vazio. Então existe um elemento com mínima norma, isto é, existe $x^* \in S^*$ tal que $\|x^*\| \leq \|x\|$ para todo $x \in S^*$.*

Demonstração. Defina $\gamma = \inf\{\|x\| \mid x \in S^*\}$. Então existe $(x^k) \subset S^*$ tal que $\|x^k\| \rightarrow \gamma$. Portanto, (x^k) é limitada e conseqüentemente admite uma subsequência convergente, seja

ela $x^k \xrightarrow{\mathbb{N}'} x^*$. Como S^* é fechado, temos que $x^* \in S^*$. Além disso, $\|x^k\| \xrightarrow{\mathbb{N}'} \|x^*\|$, donde segue que $\|x^*\| = \gamma$. \square

Agora, temos condições de provar o teorema da existência de solução global para programação quadrática.

Teorema 1.34 *Considere o problema quadrático (1.6), suponha que o seu conjunto viável seja não vazio e que a função objetivo seja limitada inferiormente neste conjunto. Então o problema tem um minimizador global.*

Demonstração. Denotamos por Ω o conjunto viável do problema (1.6). Ou seja, $\Omega = \{x \in \mathbb{R}^n \mid Ax + b \leq 0\}$. Considere um ponto qualquer $\bar{x} \in \Omega$ e defina, para cada $\rho \geq \bar{\rho} \stackrel{\text{def}}{=} \|\bar{x}\|$, o conjunto $\Omega_\rho = \{x \in \Omega \mid \|x\| \leq \rho\}$. O conjunto dos minimizadores globais da função contínua f no conjunto compacto Ω_ρ é não vazio e conseqüentemente, pelo Lema 1.33, possui um elemento de mínima norma, que será denotado por x_ρ . Note que a função $\rho \mapsto f(x_\rho)$ é não crescente e

$$\lim_{\rho \rightarrow \infty} f(x_\rho) = \inf_{x \in \Omega} f(x) \stackrel{\text{def}}{=} f^*. \quad (1.7)$$

De fato, dado $\epsilon > 0$ tome $x' \in \Omega$ tal que $f(x') < f^* + \epsilon$. Assim, para todo $\rho \geq \|x'\|$ temos $f(x_\rho) \leq f(x') \leq f^* + \epsilon$. Como $f(x_\rho) \geq f^*$, concluímos que (1.7) vale. Afirmamos que existe $\tilde{\rho} \geq \bar{\rho}$ tal que $\|x_\rho\| < \rho$ para todo $\rho \geq \tilde{\rho}$. Suponha por absurdo que isto não seja verdade. Então existe uma seqüência (ρ_k) tal que $\rho_k \rightarrow \infty$ e $\|x_{\rho_k}\| = \rho_k$. Definindo $x^k = x_{\rho_k}$ e $u^k = \frac{x^k}{\rho_k}$, podemos sem perda de generalidade supor que $u^k \rightarrow u$, com $\|u\| = 1$.

Denote as linhas da matriz A por a_i^T , $i \in I \stackrel{\text{def}}{=} \{1, \dots, m\}$ e considere o conjunto

$$I_0 = \{i \in I \mid \limsup_{k \rightarrow \infty} a_i^T x^k + b_i = 0\}.$$

Como $Ax^k + b \leq 0$, podemos concluir que $\limsup_{k \rightarrow \infty} a_i^T x^k + b_i < 0$ para $i \in I \setminus I_0$. Portanto, descartando (se necessário) uma quantidade finita de termos da seqüência (x^k) , podemos concluir que existe $\epsilon > 0$ tal que

$$a_i^T x^k + b_i \leq -\epsilon, \quad (1.8)$$

para todo $i \in I \setminus I_0$ e para todo $k \in \mathbb{N}$. Note que $a_i^T x^k + b_i = \rho_k a_i^T u^k + b_i$ e $a_i^T u^k \rightarrow a_i^T u$. Como $\rho_k \rightarrow \infty$, podemos concluir que

$$a_i^T u = 0, \quad i \in I_0 \quad \text{e} \quad a_i^T u \leq 0, \quad i \in I \setminus I_0. \quad (1.9)$$

De (1.7) obtemos

$$\rho_k (\rho_k (u^k)^T H u^k + c^T u^k) = (x^k)^T H x^k + c^T x^k = f(x^k) \rightarrow f^*$$

e portanto

$$u^T H u = 0. \quad (1.10)$$

Usando (1.9), obtemos $x^k + \lambda u \in \Omega$ para todo $k \in \mathbb{N}$ e para todo $\lambda \geq 0$. Além disso, em vista de (1.10),

$$f(x^k + \lambda u) = f(x^k) + \lambda(c^T u + 2u^T H x^k),$$

donde segue que $c^T u + 2u^T H x^k \geq 0$ (pois do contrário teríamos $\lim_{\lambda \rightarrow \infty} f(x^k + \lambda u) = -\infty$). Desta forma obtemos

$$f(x^k - \lambda u) = f(x^k) - \lambda(c^T u + 2u^T H x^k) \leq f(x^k), \quad (1.11)$$

para todo $k \in \mathbb{N}$ e para todo $\lambda \geq 0$. Mostraremos que existem $\lambda \geq 0$ e $k \in \mathbb{N}$ tais que $x^k - \lambda u \in \Omega_{\rho_k}$ e $\|x^k - \lambda u\| < \|x^k\|$. De fato, por (1.9), obtemos para $i \in I_0$,

$$a_i^T(x^k - \lambda u) + b_i = a_i^T x^k + b_i \leq 0.$$

Por outro lado, para $i \in I \setminus I_0$, segue de (1.8) que

$$a_i^T(x^k - \lambda u) + b_i = a_i^T x^k + b_i - \lambda a_i^T u \leq -\varepsilon - \lambda a_i^T u \leq 0, \quad (1.12)$$

para todo $\lambda \geq 0$ suficientemente pequeno. Agora, lembrando que $x^k = \rho_k u^k$ e $\|u\| = 1$, temos

$$\|x^k - \lambda u\|^2 = \|x^k\|^2 + \lambda(\lambda - 2\rho_k u^T u^k).$$

Escolhendo $\lambda > 0$ satisfazendo (1.12) e $k \in \mathbb{N}$ suficientemente grande de modo que $\lambda - 2\rho_k u^T u^k < 0$, podemos concluir que $\|x^k - \lambda u\| < \|x^k\| = \rho_k$. Mas isto junto com (1.11) contradiz o fato de x^k ser minimizador de f em Ω_{ρ_k} com mínima norma. Esta contradição prova a nossa afirmação, ou seja, existe $\tilde{\rho} \geq \bar{\rho}$ tal que $\|x_\rho\| < \rho$ para todo $\rho \geq \tilde{\rho}$. Para finalizar a demonstração do teorema, mostraremos que existe $\rho \geq \tilde{\rho}$ tal que $f(x_\rho) = f^*$. Suponha que isto não ocorra. Então $f(x_\rho) > f^*$ para todo $\rho \geq \tilde{\rho}$. Considere $\rho_1 \geq \tilde{\rho}$ arbitrário. Como $\lim_{\rho \rightarrow \infty} f(x_\rho) = f^*$, existe $\rho_2 > \rho_1$ tal que $f(x_{\rho_2}) < f(x_{\rho_1})$. Assim, $x_{\rho_2} \notin \Omega_{\rho_1}$ e portanto

$$\rho_1 < \|x_{\rho_2}\| < \rho_2.$$

Definindo $\hat{\rho} = \|x_{\rho_2}\|$, temos que $f(x_{\rho_2}) \leq f(x_{\hat{\rho}})$ (pela monotonicidade) e $f(x_{\hat{\rho}}) \leq f(x_{\rho_2})$ (pois $x_{\rho_2} \in \Omega_{\hat{\rho}}$). Além disso, como $\hat{\rho} \geq \tilde{\rho}$, temos $\|x_{\hat{\rho}}\| < \hat{\rho} = \|x_{\rho_2}\|$, contradizendo o fato de x_{ρ_2} ser minimizador de mínima norma em Ω_{ρ_2} . \square

1.2.3 Dualidade

A teoria de dualidade baseia-se em associar um problema de minimização (primal) a um problema de maximização (dual), que sob certas condições é equivalente ao primal

e pode ter uma resolução mais fácil. Além disso, quando o problema primal é convexo, obtemos relações de dualidade mais fortes.

Em problemas com restrições, é conveniente definir uma função, conhecida como Lagrangiana, que incorpora informações sobre a função objetivo e as restrições. Para melhor compreensão podemos consultar [11, 17].

Definição 1.35 *Dado um problema de Otimização, como o descrito em (1.3), definimos a função Lagrangiana $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ por*

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^m \alpha_i g_i(x) + \sum_{i=1}^p \beta_i h_i(x) = f(x) + \alpha^T g(x) + \beta^T h(x),$$

onde os coeficientes α_i e β_i são chamados de multiplicadores de Lagrange.

O problema dual de (1.3) é definido por

$$\begin{aligned} \max_{\alpha, \beta} \quad & \theta(\alpha, \beta) \\ \text{s.a.} \quad & \alpha \geq 0, \end{aligned} \tag{1.13}$$

onde $\theta : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R} \cup \{-\infty\}$, chamada função dual, é dada por $\theta(\alpha, \beta) = \inf_{x \in \mathbb{R}^n} L(x, \alpha, \beta)$.

Note que o problema dual (1.13) é equivalente ao problema

$$\begin{aligned} \min_{\alpha, \beta} \quad & -\theta(\alpha, \beta) \\ \text{s.a.} \quad & (\alpha, \beta) \in D, \end{aligned} \tag{1.14}$$

onde

$$D = \{(\alpha, \beta) \in \mathbb{R}^m \times \mathbb{R}^p \mid \alpha \geq 0 \text{ e } \theta(\alpha, \beta) > -\infty\}. \tag{1.15}$$

Uma propriedade interessante é que, mesmo o problema primal não cumprindo nenhuma hipótese de convexidade, o dual será sempre equivalente a um problema convexo.

Teorema 1.36 *O conjunto definido em (1.15) é convexo e a função dual θ é côncava, o que significa que o problema (1.14) é convexo.*

Demonstração. Dados $x \in \mathbb{R}^n$, $(\alpha_1, \beta_1), (\alpha_2, \beta_2) \in D$ e $t \in [0, 1]$, defina

$$(\alpha, \beta) = (1 - t)(\alpha_1, \beta_1) + t(\alpha_2, \beta_2).$$

Assim, $\alpha \geq 0$ e

$$L(x, \alpha, \beta) = f(x) + \alpha^T g(x) + \beta^T h(x) = (1 - t)L(x, \alpha_1, \beta_1) + tL(x, \alpha_2, \beta_2).$$

Portanto, $L(x, \alpha, \beta) \geq (1 - t)\theta(\alpha_1, \beta_1) + t\theta(\alpha_2, \beta_2)$, donde segue que

$$\theta(\alpha, \beta) = \inf_{x \in \mathbb{R}^n} L(x, \alpha, \beta) \geq (1 - t)\theta(\alpha_1, \beta_1) + t\theta(\alpha_2, \beta_2).$$

Assim, $(\alpha, \beta) \in D$ e θ é côncava, provando que o problema (1.14) é convexo. \square

O teorema seguinte estabelece que o valor da função dual nunca supera o valor da função primal em pontos viáveis para os problemas (1.3) e (1.13).

Teorema 1.37 (Dualidade fraca) *Dados $\bar{x} \in \Omega$ e $(\bar{\alpha}, \bar{\beta}) \in D$, temos $\theta(\bar{\alpha}, \bar{\beta}) \leq f(\bar{x})$.*

Demonstração.

$$\theta(\bar{\alpha}, \bar{\beta}) = \inf_{x \in \mathbb{R}^n} L(x, \bar{\alpha}, \bar{\beta}) \leq L(\bar{x}, \bar{\alpha}, \bar{\beta}) = f(\bar{x}) + \bar{\alpha}^T g(\bar{x}) \leq f(\bar{x}).$$

\square

Agora, veremos que se ocorrer a igualdade no Teorema 1.37, então os pontos são soluções dos seus respectivos problemas.

Teorema 1.38 (Dualidade forte) *Sejam $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ funções continuamente diferenciáveis. Se $x^* \in \Omega$ e $(\alpha^*, \beta^*) \in D$ satisfazem a igualdade $\theta(\alpha^*, \beta^*) = f(x^*)$, então x^* é minimizador global do problema primal (1.3) e (α^*, β^*) é maximizador global do problema dual (1.13). Além disso, (x^*, α^*, β^*) satisfaz as condições de KKT dadas em (1.4a)–(1.4d).*

Demonstração. Note primeiro que pelo Teorema 1.37, temos

$$\theta(\alpha, \beta) \leq f(x^*) = \theta(\alpha^*, \beta^*) \leq f(x),$$

para todos $x \in \Omega$ e $(\alpha, \beta) \in D$, donde seguem as duas afirmações de otimalidade. Agora, veja que (1.4b) e (1.4c) são imediatas. Além disso, temos

$$f(x^*) = \theta(\alpha^*, \beta^*) = \inf_{x \in \mathbb{R}^n} L(x, \alpha^*, \beta^*) \leq L(x^*, \alpha^*, \beta^*) = f(x^*) + (\alpha^*)^T g(x^*) \leq f(x^*), \quad (1.16)$$

implicando em $(\alpha^*)^T g(x^*) = 0$. Como $\alpha_i^* g_i(x^*) \leq 0$, obtemos (1.4d).

De (1.16) também obtemos $\inf_{x \in \mathbb{R}^n} L(x, \alpha^*, \beta^*) = L(x^*, \alpha^*, \beta^*)$, o que implica

$$\nabla f(x^*) + \sum_{i=1}^m \alpha_i^* \nabla g_i(x^*) + \sum_{i=1}^p \beta_i^* \nabla h_i(x^*) = \nabla_x L(x^*, \alpha^*, \beta^*) = 0,$$

fornecendo (1.4a). \square

Em geral, não podemos garantir que $\theta(\alpha^*, \beta^*) = f(x^*)$. A diferença entre tais valores é definida abaixo.

Definição 1.39 *A diferença entre os valores ótimos da função objetivo do problema primal e do dual, $f(x^*) - \theta(\alpha^*, \beta^*)$, é definida como gap de dualidade [17].*

Sob hipóteses de convexidade, podemos provar que o gap de dualidade é nulo.

Teorema 1.40 *Suponha que $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, são funções continuamente diferenciáveis e convexas e que h é uma função afim. Se $x^* \in \mathbb{R}^n$ é um ponto estacionário para o problema (1.3) e $(\alpha^*, \beta^*) \in \mathbb{R}^m \times \mathbb{R}^p$ é um vetor de multiplicadores de Lagrange associado, então x^* é minimizador global do problema primal (1.3) e (α^*, β^*) é maximizador global do problema dual (1.13). Além disso, não há gap de dualidade, ou seja, $\theta(\alpha^*, \beta^*) = f(x^*)$.*

Demonstração. A otimalidade de x^* já foi provada no Teorema 1.26. Para estabelecer a otimalidade dual, note que a função $x \mapsto L(x, \alpha^*, \beta^*)$ é convexa, pois f e g_i , $i = 1, \dots, m$, são convexas, $\alpha^* \geq 0$ e h é uma função afim. Além disso, em vista de (1.4a), temos $\nabla_x L(x^*) = 0$, o que implica

$$\theta(\alpha^*, \beta^*) = \inf_{x \in \mathbb{R}^n} L(x, \alpha^*, \beta^*) = L(x^*, \alpha^*, \beta^*) = f(x^*) + (\alpha^*)^T g(x^*) + (\beta^*)^T h(x^*) = f(x^*),$$

sendo que a última igualdade decorre de (1.4d). A otimalidade global de (α^*, β^*) segue então do Teorema 1.37. \square

Exemplo 1.41 *Considere o seguinte problema de Otimização*

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & f(x) = x_1^2 + x_2^2 \\ \text{s.a} \quad & 2x_1 + x_2 \leq -4. \end{aligned}$$

Vamos encontrar uma solução deste problema por meio do seu dual. A Lagrangiana associada a este problema é a função $L : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ dada por

$$L(x, \alpha) = x_1^2 + x_2^2 + 2\alpha x_1 + \alpha x_2 + 4\alpha.$$

Note que, fixado $\alpha > 0$, a função $x \mapsto L(x, \alpha)$ é diferenciável e estritamente convexa. Portanto, a equação

$$\nabla_x L(x, \alpha) = \begin{pmatrix} 2x_1 + 2\alpha \\ 2x_2 + \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

fornece o único minimizador $\bar{x} = \begin{pmatrix} -\alpha \\ -\alpha/2 \end{pmatrix}$ e assim a função dual fica

$$\theta(\alpha) = L(\bar{x}, \alpha) = -\frac{5\alpha^2}{4} + 4\alpha.$$

O problema dual é então

$$\begin{aligned} \max_{\alpha} \quad & \theta(\alpha) = -\frac{5\alpha^2}{4} + 4\alpha \\ \text{s.a} \quad & \alpha \geq 0, \end{aligned}$$

fornecendo $\alpha^ = \frac{8}{5}$ e portanto $x^* = -\frac{1}{5} \begin{pmatrix} 8 \\ 4 \end{pmatrix}$. Além disso, o gap de dualidade é zero,*

pois

$$f(x^*) = \left(-\frac{8}{5}\right)^2 + \left(-\frac{4}{5}\right)^2 = \frac{16}{5} = \theta(\alpha^*).$$

Além do que vimos nos Teoremas 1.38 e 1.40, uma outra maneira de relacionar otimalidade com o *gap* de dualidade é por meio do conceito de ponto de sela, conforme veremos a seguir.

Definição 1.42 Um vetor $(x^*, \alpha^*, \beta^*) \in \mathbb{R}^n \times \mathbb{R}_+^m \times \mathbb{R}^p$ é dito ponto de sela da função Lagrangiana associada ao problema primal (1.3) quando

$$L(x^*, \alpha, \beta) \leq L(x^*, \alpha^*, \beta^*) \leq L(x, \alpha^*, \beta^*) \quad (1.17)$$

para todos $x \in \mathbb{R}^n$ e $(\alpha, \beta) \in \mathbb{R}_+^m \times \mathbb{R}^p$.

Teorema 1.43 Suponha que $(x^*, \alpha^*, \beta^*) \in \mathbb{R}^n \times \mathbb{R}_+^m \times \mathbb{R}^p$ é um ponto de sela da função Lagrangiana. Então, $x^* \in \Omega$, $(\alpha^*, \beta^*) \in D$ e $\theta(\alpha^*, \beta^*) = f(x^*)$.

Demonstração. Vejamos primeiro que $x^* \in \Omega$. Suponha por absurdo que $g_j(x^*) > 0$ para algum $j \in \{1, \dots, m\}$. Para cada $k \in \mathbb{N}$, defina $\alpha_j = k$, $\alpha_i = 0$, $i \neq j$ e $\beta = 0$. Assim,

$$L(x^*, \alpha, \beta) = f(x^*) + kg_j(x^*) \rightarrow \infty$$

quando $k \rightarrow \infty$, contradizendo (1.17). Suponha agora, também por absurdo, que $h_j(x^*) \neq 0$ para algum $j \in \{1, \dots, p\}$. Para cada $k \in \mathbb{N}$, defina $\beta_j = kh_j(x^*)$, $\beta_i = 0$, $i \neq j$ e $\alpha = 0$. Assim,

$$L(x^*, \alpha, \beta) = f(x^*) + k(h_j(x^*))^2 \rightarrow \infty$$

quando $k \rightarrow \infty$, novamente contradizendo (1.17). Portanto, $x^* \in \Omega$. Para ver que $(\alpha^*, \beta^*) \in D$, basta notar que (1.17) implica

$$\theta(\alpha^*, \beta^*) = \inf_{x \in \mathbb{R}^n} L(x, \alpha^*, \beta^*) \geq L(x^*, \alpha^*, \beta^*) > -\infty.$$

Finalmente, note que por (1.17) $f(x^*) = L(x^*, 0, 0) \leq L(x^*, \alpha^*, \beta^*) \leq \theta(\alpha^*, \beta^*)$ e pelo Teorema 1.37, $\theta(\alpha^*, \beta^*) \leq f(x^*)$, provando assim que $\theta(\alpha^*, \beta^*) = f(x^*)$. \square

Corolário 1.44 Se $(x^*, \alpha^*, \beta^*) \in \mathbb{R}^n \times \mathbb{R}_+^m \times \mathbb{R}^p$ é um ponto de sela da função Lagrangiana, então x^* é minimizador global do problema primal (1.3), (α^*, β^*) é maximizador global do problema dual (1.13) e (x^*, α^*, β^*) satisfaz as condições de KKT dadas em (1.4a)–(1.4d).

Demonstração. Imediata pelo Teorema 1.38. \square

A recíproca do Teorema 1.43 também é verdadeira.

Teorema 1.45 Suponha que $(x^*, \alpha^*, \beta^*) \in \Omega \times D$ satisfaz $\theta(\alpha^*, \beta^*) = f(x^*)$. Então (x^*, α^*, β^*) é um ponto de sela da função Lagrangiana.

Demonstração. Pelo Teorema 1.38, temos que x^* é minimizador global do problema primal (1.3), (α^*, β^*) é maximizador global do problema dual (1.13) e (x^*, α^*, β^*) satisfaz as condições de KKT (1.4a)–(1.4d). Portanto,

$$\begin{aligned} L(x^*, \alpha, \beta) &= f(x^*) + \alpha^T g(x^*) + \beta^T h(x^*) \\ &\leq f(x^*) \\ &= f(x^*) + (\alpha^*)^T g(x^*) + (\beta^*)^T h(x^*) \\ &= L(x^*, \alpha^*, \beta^*), \end{aligned}$$

para todo $(\alpha, \beta) \in \times \mathbb{R}_+^m \times \mathbb{R}^p$. Por outro lado,

$$L(x^*, \alpha^*, \beta^*) = f(x^*) = \theta(\alpha^*, \beta^*) \leq L(x, \alpha^*, \beta^*)$$

para todo $x \in \mathbb{R}^n$. □

Corolário 1.46 *Suponha que $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, são funções continuamente diferenciáveis e convexas e que h é uma função afim. Se $x^* \in \mathbb{R}^n$ é um ponto estacionário para o problema (1.3) e $(\alpha^*, \beta^*) \in \mathbb{R}^m \times \mathbb{R}^p$ é um vetor de multiplicadores de Lagrange associado, então (x^*, α^*, β^*) é um ponto de sela da função Lagrangiana.*

Demonstração. Imediata pelo Teorema 1.40. □

Capítulo 2

Máquinas de Vetores Suporte

Quando uma grande quantidade de dados é analisada, já não é viável uma classificação manual, pelo tempo e pela dificuldade de diferenciá-los. Assim, computacionalmente buscamos gerar um algoritmo capaz de reconhecer padrões. Neste capítulo fornecemos a modelagem matemática de uma técnica que possui tal capacidade: as Máquinas de Vetores Suporte (SVM, do inglês *Support Vector Machine*).

Para fundamentar a técnica, estabeleceremos teoremas e lemas, recorrendo ao Capítulo 1. Trabalharemos inicialmente com a classificação de dados linearmente separáveis e posteriormente a generalização para dados que não possuem esta propriedade.

2.1 Metodologia de Aprendizagem

Nesta seção, faremos uma breve revisão de literatura sobre SVM. Inicialmente, daremos uma introdução à Aprendizagem de Máquina (do inglês *Machine Learning*), para então tratar sobre SVM. Neste trabalho, usamos SVM para classificação, mas também poderia ser usada para regressão, conforme podemos encontrar em [35].

Aprendizagem de Máquina é uma das áreas que têm fornecido técnicas úteis para resolver problemas de classificação com uma grande quantidade de dados [24]. O aprendizado da máquina pode ser comparado ao dos seres humanos, no sentido que ele vem com a experiência. A máquina precisa experimentar o problema várias vezes e criar seu próprio modelo da solução. Essencialmente, isso significa reconhecer padrões. Quanto mais informação for fornecida, ou seja, quanto mais experiência obter, melhor vai ser o aprendizado, sendo mais fácil de reconhecer o padrão.

Por exemplo, dada uma sequência, tentaremos descobrir o próximo número. Se for uma sequência de um ou dois números, por exemplo $(3, 6, \dots)$, dificilmente saberemos qual é a regra que gera esta sequência e assim conseguir obter o próximo valor. Mas, quanto mais números obtivermos da sequência, mais fácil é encontrar o padrão, como $(3, 6, 9, 12, \dots)$, onde o padrão é uma progressão aritmética de razão 3. Este é um tipo de aprendizado, o supervisionado, onde temos informação sobre alguns dados e a partir

disso podemos deduzir uma informação sobre um caso desconhecido. Neste trabalho abordaremos o aprendizado supervisionado.

Esse tipo de aprendizado é muito usado, por exemplo, em problema de reconhecimento facial. Suponha que a máquina precise reconhecer uma pessoa. Precisamos alimentar a máquina com várias fotos e colocar um rótulo nessas imagens, dizendo para ela quais imagens são e quais não são a pessoa.

Assim, a partir de cada uma daquelas fotos, criamos um modelo de características do rosto da pessoa. Feito isso, toda vez que for apresentada uma nova imagem, basta comparar o modelo de características que a máquina criou a partir dos outros dados com as características da nova imagem. Se as características forem semelhantes, significa que provavelmente essa nova imagem é a pessoa.

Muitas técnicas são estudadas para Aprendizagem de Máquina, tanto para a aprendizagem supervisionada, como SVM, Regressão Linear, Regressão Logística e Redes Neurais [3], quanto para a não supervisionada como SVD (*Singular Value Decomposition*) [13], Clusterização e Análise de Componentes Principais (do inglês *Principal Component Analysis*) [3].

O objetivo das técnicas de aprendizagem supervisionada é ter a capacidade de obter conhecimento, em um processo denominado fase de treinamento, de vários pares $(x^1, y_1), (x^2, y_2), \dots, (x^m, y_m)$, de entrada x^i e saída y_i , denominado conjunto de treinamento. Para analisar se a técnica obteve conhecimento suficiente, testamos novas entradas para as quais já conhecemos a priori as saídas, denominado conjunto de teste, verificando se a máquina gera saídas corretas. Este processo é conhecido como fase de teste. Dito de outro modo, queremos prever a saída para dados novos através de um classificador, uma função de decisão.

A técnica SVM está entre uma das mais utilizadas em aprendizado supervisionado [4, 5, 15, 21, 28]. Tem boa capacidade de generalização, robustez diante de dados em dimensões elevadas e tem sido muito utilizada por obter resultados melhores em relação a outras técnicas de classificação [7, 30, 32, 33]. Seu objetivo é encontrar um hiperplano ótimo $(w^*)^T x + b^* = 0$ que separe os dados de entrada x^i em duas saídas (classes) y_i através de uma função de decisão $F(x) = \text{sign}((w^*)^T x + b^*)$. A Otimização fornece técnicas para encontrar tal hiperplano, garantindo não somente sua existência mas também sua unicidade.

Considere um conjunto de dados, pertencentes a duas classes, conforme Figura 2.1. Observe que só podemos classificar corretamente os dados através de uma reta na Figura 2.1(a). Na Figura 2.1(b) encontramos uma reta que classifica alguns poucos dados incorretamente. Por fim, na Figura 2.1(c) não existe uma reta que separe os dados de maneira razoável como aconteceu nos dois primeiros casos. Esta é a motivação para tratarmos dos três casos de SVM: o linear com margem rígida, restrito a poucas aplicações práticas, o linear com margem flexível e o não linear, respectivamente.

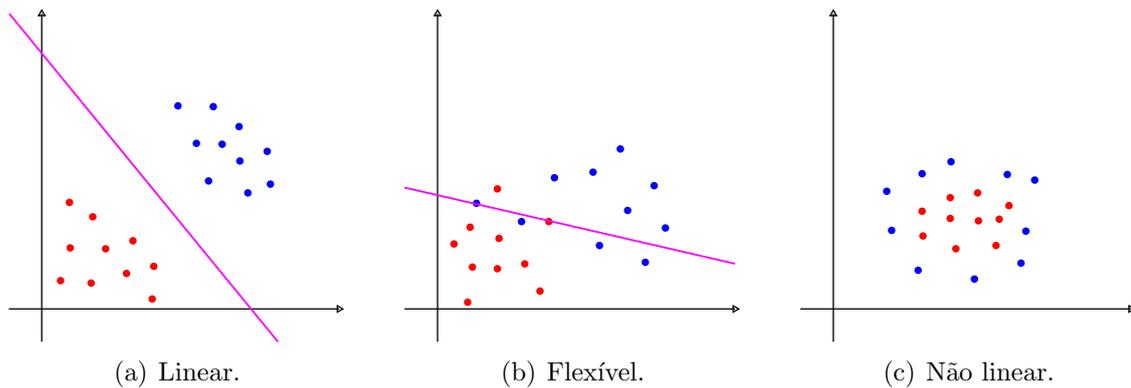


Figura 2.1: Dados lineares, com margem flexível e não lineares.

Em 1963, Vapnik [34] apresentou um algoritmo para encontrar um hiperplano separador para o caso em que os dados são linearmente separáveis. Depois, em 1992, Boser, Guyon e Vapnik [22] trabalharam com classificação não linear, aplicando o truque do *Kernel*, que faz um mapeamento do espaço de entrada para um espaço de dimensão maior, no qual os dados são linearmente separáveis. Em 1995, Cortes e Vapnik [21] modificaram o algoritmo, introduzindo variáveis de folga para o caso em que os dados não são linearmente separáveis.

No decorrer do texto, para os três casos descritos acima, consideraremos o conjunto de treinamento $X = \{(x^1, y_1), \dots, (x^m, y_m), x^i \in \mathbb{R}^n, y_i \in \{-1, 1\}\}$, pois trataremos de problemas de classificação binária. Denominaremos atributo aos pontos x^i , pertencentes ao espaço de entrada. Quando a saída $y_i = 1$ diremos que x^i pertence à classe positiva e se $y_i = -1$ diremos que x^i pertence à classe negativa.

Por exemplo, considere um conjunto de m letras, formado por tipos distintos da letra A e de letras que não são A. A cada letra associamos uma matriz de pixels $p \times q$, a qual produz um vetor de características $x^i \in \mathbb{R}^{pq} = \mathbb{R}^n$, como representado na Figura 2.2. Os dois primeiros caracteres têm saída $y_i = 1$, enquanto que o último tem saída $y_i = -1$.

Mostraremos que a modelagem do problema de classificação, por meio de SVM, envolve um problema de programação não linear, quadrático, convexo e restrito, no qual almejamos obter a maior distância entre os hiperplanos $w^T x + b = 1$ e $w^T x + b = -1$ que definem a faixa que separa os dados.

Usaremos a formulação dual Lagrangiana para resolver este problema de otimização. Em [4], encontramos duas razões para fazer isso, uma delas é que as restrições do problema primal são trocadas por restrições mais simples na abordagem dual e a outra é que os dados aparecem em forma de produto interno, propriedade fundamental para poder generalizar SVM para o caso não linear. Além dessas razões, também fornecemos neste trabalho uma fundamentação teórica para outros motivos de se utilizar o dual. Detalhes sobre dualidade no contexto de SVM podem ser encontrados nos trabalhos de Nello Cristianini e John Shawe-Taylor [15, 16].

$$\begin{array}{c}
 \mathbf{A} \rightarrow \begin{pmatrix} a_{11} & \cdots & a_{1q} \\ \vdots & \ddots & \vdots \\ a_{p1} & \cdots & a_{pq} \end{pmatrix}_{p \times q} \rightarrow \begin{pmatrix} a_1 \\ \vdots \\ a_{pq} \end{pmatrix} \in \mathbb{R}^{pq} = \mathbb{R}^n. \\
 \\
 \mathbf{a} \rightarrow \begin{pmatrix} \tilde{a}_{11} & \cdots & \tilde{a}_{1q} \\ \vdots & \ddots & \vdots \\ \tilde{a}_{p1} & \cdots & \tilde{a}_{pq} \end{pmatrix}_{p \times q} \rightarrow \begin{pmatrix} \tilde{a}_1 \\ \vdots \\ \tilde{a}_{pq} \end{pmatrix} \in \mathbb{R}^{pq} = \mathbb{R}^n. \\
 \\
 \mathbf{B} \rightarrow \begin{pmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pq} \end{pmatrix}_{p \times q} \rightarrow \begin{pmatrix} b_1 \\ \vdots \\ b_{pq} \end{pmatrix} \in \mathbb{R}^{pq} = \mathbb{R}^n.
 \end{array}$$

Figura 2.2: Transformação de caractere em vetor.

Neste trabalho, apresentamos uma fundamentação teórica da existência e unicidade da solução do problema primal. Por outro lado, não podemos garantir que o problema dual tem solução única e mostramos um exemplo para o qual o dual possui infinitas soluções. Além disso, mostraremos que o *gap* de dualidade é zero, o que permite estabelecer uma relação entre uma solução qualquer do dual e a solução do primal. Por conta desta relação, é interessante obter uma solução do dual com o maior número possível de componentes nulas, pois isto diminui o custo computacional para obter a função de decisão e também para classificar novos pontos.

Existem trabalhos na literatura [2, 15, 28] que definem vetores suporte como os pontos do conjunto de treinamento que estão sobre os hiperplanos que determinam a margem ótima. Outros trabalhos, no entanto, definem vetores suporte como os atributos do conjunto de treinamento associados às componentes não nulas de uma solução do problema dual [35]. Baseados nesta última definição, propomos neste trabalho o Algoritmo 2.1, que busca uma solução dual o mais esparsa possível.

Quando o conjunto de dados não é linearmente separável, uma forma de impedir classificações incorretas é utilizar regularização, introduzindo variáveis de folga ξ_i associadas a cada atributo x^i , penalizadas na função objetivo por meio de um parâmetro $C > 0$. Em [2] é realizado um estudo da influência do parâmetro C no desempenho dos classificadores. Faremos alguns testes no Capítulo 3 para verificação de tal relação e analisaremos quão grande deve ser o parâmetro C para obtermos menos erros na classificação.

Para o caso de aplicarmos as variáveis de folga e ainda assim obtermos uma baixa taxa de acerto na classificação, podemos fazer um mapeamento dos dados para um espaço de dimensão maior por meio de uma função $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^N$ com $N > n$, na esperança de

que neste novo espaço os dados sejam linearmente separáveis. Isto não acarreta grandes complicações na modelagem, pois na função objetivo do problema dual aparece apenas uma expressão do tipo $K(x^i, x^j) = \phi(x^i)^T \phi(x^j)$, denominada função *Kernel*, no lugar de $(x^i)^T x^j$. O *Kernel* também aparece na função de decisão, o que significa que mesmo sendo o hiperplano de separação criado no espaço \mathbb{R}^N , a avaliação de novos pontos se dá no espaço de entrada [5].

A seguir, veremos o desenvolvimento de SVM com dados linearmente separáveis e nas seções seguintes suas generalizações.

2.2 Máquinas de Vetores Suporte - Margem rígida

Primeiramente, trataremos sobre o problema com margem rígida, ou seja, quando os dados são linearmente separáveis. Para tanto, vamos definir alguns conceitos sobre hiperplano e formular o problema com o objetivo de separar os dados da melhor maneira. Usaremos conceitos de dualidade para resolver o problema de otimização e encontrar o classificador ótimo.

Considere a Figura 2.3(a), na qual os pontos em azul representam dados de treinamento da classe positiva e os em vermelho dados de treinamento da classe negativa. Note que todos os hiperplanos representados na Figura 2.3(b) separam os dados. No entanto, dentre todos os hiperplanos separadores, aquele que está destacado na cor magenta, na Figura 2.3(c), é o que melhor separa os dados pois possibilita a maior faixa que não contém nenhum atributo, conforme Figura 2.3(d). De fato, se o hiperplano tiver uma margem estreita, então pequenas perturbações no hiperplano ou no conjunto de dados podem ocasionar uma classificação incorreta.

Vamos agora formalizar estas ideias com o objetivo de modelar e formular matematicamente o problema de classificação. Considere os conjuntos

$$X_1 = \{x^i \in X \mid y_i = 1\} \quad \text{e} \quad X_2 = \{x^i \in X \mid y_i = -1\},$$

formados pelos atributos pertencentes às classes positiva e negativa, respectivamente.

Definição 2.1 Considere um vetor não nulo $w \in \mathbb{R}^n$ e um escalar $b \in \mathbb{R}$. Um hiperplano com vetor normal w e constante b é um conjunto da forma $H = \{x \in \mathbb{R}^n \mid w^T x + b = 0\}$.

O hiperplano H divide o espaço \mathbb{R}^n em dois semiespaços, dados por

$$S_1 = \{x \in \mathbb{R}^n \mid w^T x + b \geq 0\} \quad \text{e} \quad S_2 = \{x \in \mathbb{R}^n \mid w^T x + b \leq 0\}.$$

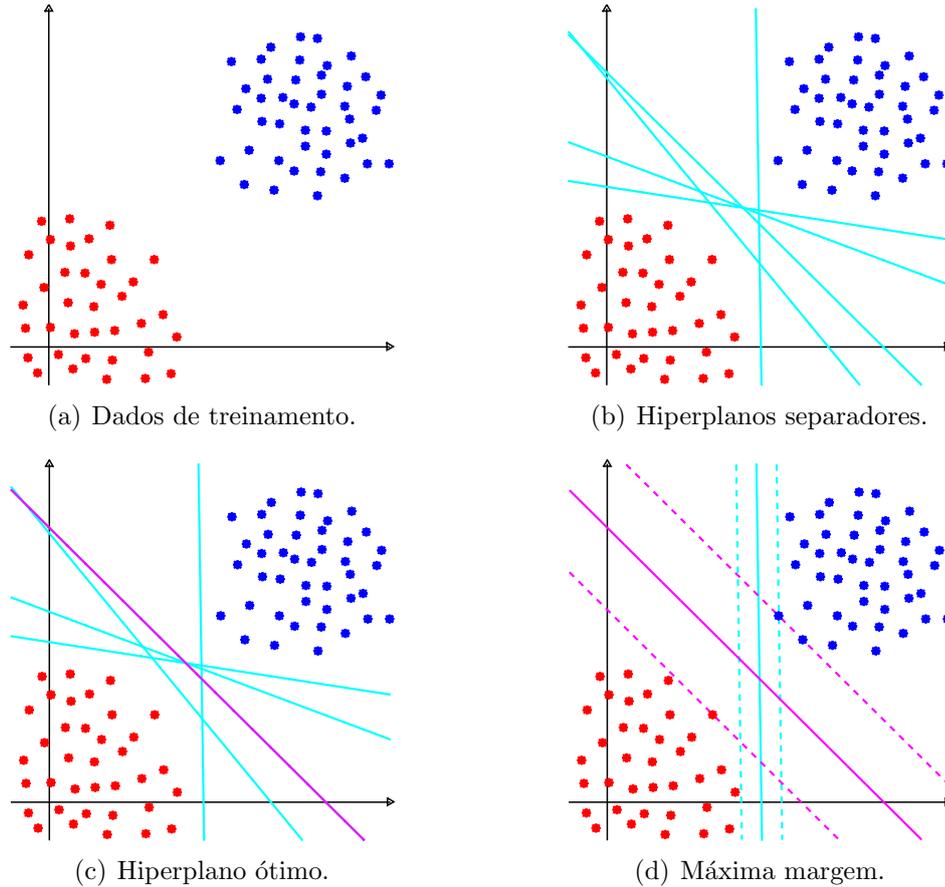


Figura 2.3: Hiperplanos.

Definição 2.2 Dizemos que os conjuntos $X_1, X_2 \subset \mathbb{R}^n$ são linearmente separáveis quando existem $w \in \mathbb{R}^n$ e $b \in \mathbb{R}$ tais que $w^T x + b > 0$ para todo $x \in X_1$ e $w^T x + b < 0$ para todo $x \in X_2$. O hiperplano $H = \{x \in \mathbb{R}^n \mid w^T x + b = 0\}$ é chamado hiperplano separador dos conjuntos X_1 e X_2 .

Lema 2.3 Suponha que os conjuntos $X_1, X_2 \subset \mathbb{R}^n$ são finitos e linearmente separáveis, com hiperplano separador $H = \{x \in \mathbb{R}^n \mid w^T x + b = 0\}$. Então existem $\bar{w} \in \mathbb{R}^n$ e $\bar{b} \in \mathbb{R}$ tais que H pode ser descrito por

$$\bar{w}^T x + \bar{b} = 0,$$

satisfazendo

$$\bar{w}^T x + \bar{b} \geq 1, \text{ para todo } x \in X_1 \text{ e}$$

$$\bar{w}^T x + \bar{b} \leq -1, \text{ para todo } x \in X_2.$$

Demonstração. Pela Definição 2.2, temos que

$$w^T x + b > 0, \text{ para todo } x \in X_1$$

$$w^T x + b < 0, \text{ para todo } x \in X_2.$$

Como $X_1 \cup X_2$ é um conjunto finito, $\gamma \stackrel{\text{def}}{=} \min_{x \in X_1 \cup X_2} |w^T x + b| > 0$. Portanto, para todo

$x \in X_1 \cup X_2$, $\gamma \leq |w^T x + b|$ e, conseqüentemente, $\frac{|w^T x + b|}{\gamma} \geq 1$. Assim, para $x \in X_1$ temos

$$\frac{w^T x + b}{\gamma} = \frac{|w^T x + b|}{\gamma} \geq 1$$

e para $x \in X_2$ temos

$$-\frac{w^T x + b}{\gamma} = \frac{|w^T x + b|}{\gamma} \geq 1.$$

Definindo $\bar{w} = \frac{w}{\gamma}$ e $\bar{b} = \frac{b}{\gamma}$, obtemos o resultado. \square

Para fins de notação, vamos considerar a partir de agora (w, b) ao invés de (\bar{w}, \bar{b}) . Veremos que $H_1 \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid w^T x + b = 1\}$ e $H_2 \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid w^T x + b = -1\}$ são os hiperplanos que definem a faixa que separa os conjuntos X_1 e X_2 . No lema abaixo estabelecemos a largura desta faixa.

Lema 2.4 *A distância entre os hiperplanos H_1 e H_2 é dada por $d = \frac{2}{\|w\|}$.*

Demonstração. Considere um ponto arbitrário $\bar{x} \in H_1$ e defina $x^* = \bar{x} + \lambda w$, com $\lambda = -\frac{w^T \bar{x} + b + 1}{\|w\|^2}$. Note primeiro que $x^* \in H_2$. Além disso, temos que $\|\bar{x} - x^*\|_2 \leq \|\bar{x} - x\|_2$ para todo $x \in H_2$. De fato, basta tomar $u = \bar{x} - x^*$ e $v = x^* - x$. Assim $u^T v = -\lambda w^T (x^* - x) = 0$ e portanto temos $\|u\|^2 + \|v\|^2 = \|u + v\|^2$, ou seja, $\|\bar{x} - x^*\|^2 + \|x^* - x\|^2 = \|\bar{x} - x\|^2$. Portanto,

$$d(\bar{x}, H_2) = \|\bar{x} - x^*\| = |\lambda| \|w\| = \frac{|w^T \bar{x} + b + 1|}{\|w\|} = \frac{2}{\|w\|},$$

independentemente do ponto $\bar{x} \in H_1$. Deste modo, temos que

$$d(H_1, H_2) = \frac{2}{\|w\|}.$$

\square

Como o nosso objetivo é encontrar o hiperplano que melhor separa os dados, precisamos maximizar a largura da faixa, isto é, maximizar a margem $d = \frac{2}{\|w\|}$. Isto equivale a minimizar seu inverso $\frac{1}{2}\|w\|$ ou ainda minimizar $\frac{1}{2}\|w\|^2$. Além disso, a faixa deve separar os dados das duas classes. Isto significa que as seguintes restrições devem ser satisfeitas

$$w^T x + b \geq 1, \text{ para todo } x \in X_1 \text{ e} \quad (2.1)$$

$$w^T x + b \leq -1, \text{ para todo } x \in X_2. \quad (2.2)$$

De uma forma compacta, podemos reescrever as restrições acima como

$$y_i(w^T x^i + b) \geq 1, \quad i = 1, \dots, m.$$

Portanto, o problema de encontrar o hiperplano ótimo pode ser formulado da

seguinte maneira

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.a} \quad & y_i(w^T x^i + b) \geq 1, \quad i = 1, \dots, m, \end{aligned} \quad (2.3)$$

onde $w \in \mathbb{R}^n$ e $b \in \mathbb{R}$. O problema (2.3) possui função objetivo

$$f(w, b) = \frac{1}{2} \|w\|^2 \quad (2.4)$$

convexa e restrições lineares

$$g_i(w, b) = 1 - y_i(w^T x^i + b) \leq 0, \quad i = 1, \dots, m. \quad (2.5)$$

Mas, como tratado no Capítulo 1, estas hipóteses não são suficientes para garantir a existência de um minimizador. No entanto, como a função objetivo é limitada inferiormente e o conjunto viável $\Omega = \{(w, b) \in \mathbb{R}^{n+1} \mid g_i(w, b) \leq 0, i = 1, \dots, m\}$ é um poliedro não vazio (pois os conjuntos X_1 e X_2 são linearmente separáveis), o Teorema 1.34 garante a existência de um minimizador global para o problema (2.3). Ainda assim, apresentamos aqui uma demonstração da existência de um minimizador global que faz uso de conceitos mais elementares do que aqueles usados na prova do Teorema 1.34.

Teorema 2.5 *O problema (2.3) possui um minimizador global.*

Demonstração. Como a função objetivo do problema (2.3) é limitada inferiormente e o conjunto $\Omega \neq \emptyset$, temos

$$f^* \stackrel{\text{def}}{=} \inf_{(w,b) \in \Omega} f(w, b) > -\infty.$$

Pela definição de ínfimo, temos que para todo $k \in \mathbb{N}$, existe $(w_k, b_k) \in \Omega$ tal que

$$f^* \leq f(w_k, b_k) < f^* + \frac{1}{k},$$

deste modo,

$$\frac{1}{2} \|w_k\|^2 = f(w_k, b_k) \rightarrow f^* \quad (2.6)$$

e portanto $(\|w_k\|)$ é convergente, conseqüentemente pelo Teorema 1.2 é limitada, implicando que (w_k) seja limitada.

Podemos mostrar que (b_k) também será limitada. De fato, considere $\tilde{x} \in X_1$ e $\bar{x} \in X_2$ arbitrários. Portanto, por (2.1) e (2.2) temos que

$$1 - w_k^T \tilde{x} \leq b_k \leq -1 - w_k^T \bar{x}.$$

Então (b_k) é limitada, pois (w_k) é limitada.

Como a seqüência (w_k, b_k) é limitada, pelo Teorema 1.3 possui uma subsequência

convergente, seja ela $(w_k, b_k) \xrightarrow{\mathbb{N}'} (w^*, b^*)$. Logo, pela continuidade de f obtemos

$$f(w_k, b_k) \xrightarrow{\mathbb{N}'} f(w^*, b^*),$$

mas por (2.6) temos $f(w^*, b^*) = f^*$. Portanto, provamos que existe (w^*, b^*) em Ω tal que

$$f(w^*, b^*) = f^* \leq f(w, b), \text{ para todo } (w, b) \in \Omega.$$

□

Para o problema quadrático (1.6) o Teorema 1.34 garante a existência de uma solução global, mas não se pode afirmar nada sobre a unicidade desta solução. No entanto, para o problema (2.3) podemos garantir a unicidade da solução. Para mostrar este resultado, precisamos do seguinte lema.

Lema 2.6 *Se (w^*, b^*) é ótimo para o problema (2.3) então existem $\bar{x} \in X_1$ e $\tilde{x} \in X_2$ tais que $(w^*)^T \bar{x} + b^* = 1$ e $(w^*)^T \tilde{x} + b^* = -1$.*

Demonstração. Como (w^*, b^*) é ótimo para o problema (2.3), então $f(w^*, b^*) \leq f(w, b)$ para todo $(w, b) \in \Omega$. Suponha que não existe $\bar{x} \in X_1$ satisfazendo $(w^*)^T \bar{x} + b^* = 1$. Logo, $(w^*)^T x + b^* \geq 1 + \delta$ para todo $x \in X_1$, onde $\delta = \min_{x \in X_1} [(w^*)^T x + b^* - 1] > 0$. Assim,

$$(w^*)^T x + b^* - \frac{\delta}{2} \geq 1 + \frac{\delta}{2},$$

o que equivale a

$$\frac{(w^*)^T x + b^* - \frac{\delta}{2}}{1 + \frac{\delta}{2}} \geq 1.$$

Tomando $\bar{w} = \frac{w^*}{1 + \frac{\delta}{2}}$ e $\bar{b} = \frac{b^* - \frac{\delta}{2}}{1 + \frac{\delta}{2}}$, temos

$$\bar{w}^T x + \bar{b} \geq 1,$$

para todo $x \in X_1$.

Além disso, para $x \in X_2$ temos $(w^*)^T x + b^* \leq -1$ o que significa

$$(w^*)^T x + b^* - \frac{\delta}{2} \leq -1 - \frac{\delta}{2}.$$

Portanto, $\bar{w}^T x + \bar{b} \leq -1$ para todo $x \in X_2$.

Então encontramos (\bar{w}, \bar{b}) satisfazendo as restrições do problema (2.3) e

$$f(\bar{w}, \bar{b}) = \frac{1}{2} \|\bar{w}\|^2 = \frac{1}{2} \frac{\|w^*\|^2}{\left(1 + \frac{\delta}{2}\right)^2} < \frac{1}{2} \|w^*\|^2 = f(w^*, b^*),$$

o que é uma contradição pois (w^*, b^*) é ótimo. \square

Teorema 2.7 *O minimizador global para o problema (2.3) é único.*

Demonstração. Suponha que (\bar{w}, \bar{b}) e (\tilde{w}, \tilde{b}) são soluções ótimas para o problema (2.3). Então $\frac{1}{2} \|\bar{w}\|^2 = \frac{1}{2} \|\tilde{w}\|^2$ e conseqüentemente $\|\bar{w}\| = \|\tilde{w}\|$. Como Ω é um conjunto convexo, temos que $(\hat{w}, \hat{b}) = \frac{1}{2} \left((\bar{w}, \bar{b}) + (\tilde{w}, \tilde{b}) \right)$ pertence a Ω . Além disso, pelo Lema 1.1 temos que $\|\hat{w}\| < \|\bar{w}\|$ o que implica que $f(\hat{w}, \hat{b}) < f(\bar{w}, \bar{b})$, contradizendo o fato de (\bar{w}, \bar{b}) ser uma solução ótima. Portanto, temos a unicidade de w .

Agora, mostraremos a unicidade de b . Suponha que (w, \bar{b}) e (w, \tilde{b}) são soluções ótimas para o problema (2.3) com, por exemplo, $\tilde{b} < \bar{b}$. Pelo Lema 2.6, existe $x \in X_1$ tal que $w^T x + \bar{b} = 1$. Por outro lado,

$$1 \leq w^T x + \tilde{b} < w^T x + \bar{b} = 1,$$

o que é uma contradição. \square

Assim, depois de resolvido o problema primal (2.3) e encontrados w^* e b^* , podemos classificar um novo dado $x \in \mathbb{R}^n$ usando a função de decisão $F : \mathbb{R}^n \rightarrow \mathbb{R}$ dada por

$$F(x) = \text{sign} \left((w^*)^T x + b^* \right). \quad (2.7)$$

Caso $F(x) > 0$, o ponto x será classificado como da classe positiva e se $F(x) < 0$, o ponto x será classificado como da classe negativa. Mais do que isso, note que o hiperplano separador é a curva de nível 0 da função $G : \mathbb{R}^n \rightarrow \mathbb{R}$ dada por $G(x) = (w^*)^T x + b^*$. Assim, podemos considerar o valor absoluto de $G(x)$: caso $|G(x)| \gg 0$ podemos afirmar com um nível maior de confiança que este ponto pertence a classe atribuída do que se $|G(x)| \approx 0$. Isto pode ser ilustrado na Figura 2.4, na qual os pontos A e B pertencem à classe azul, mas temos mais segurança nesta classificação para o ponto A do que para o ponto B .

Vimos então que o problema de classificação pode ser resolvido por meio do problema primal (2.3). Entretanto, considere a situação ilustrada na Figura 2.5. Ao resolver o problema primal associado, encontramos a solução (w^*, b^*) . Note que, apenas três pontos definem completamente o hiperplano separador. Todos os outros pontos, que estão associados a restrições inativas, são desnecessários para a obtenção da solução. Isto será provado no Teorema 2.8. A formulação primal só detecta os pontos desnecessários após resolver o problema primal e avaliar todas as restrições na solução, enquanto a

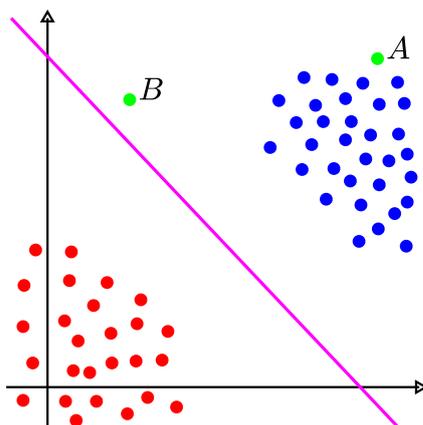


Figura 2.4: Separação de dados.

formulação dual consegue detectar isso apenas com a positividade da sua solução, como mostraremos na sequência do texto.

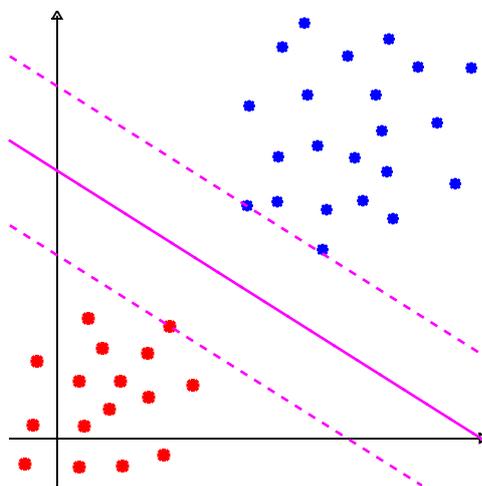


Figura 2.5: Dados sobre os hiperplanos tracejados correspondem as restrições ativas.

Teorema 2.8 Considere (w^*, b^*) a solução do problema (2.3) e $I^* = I(w^*, b^*)$ o conjunto dos índices das restrições ativas na solução. Então a solução do problema

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.a.} \quad & y_i(w^T x^i + b) \geq 1, \quad i \in I^* \end{aligned} \tag{2.8}$$

é (w^*, b^*) .

Demonstração. Seja (\bar{w}, \bar{b}) a solução do problema (2.8). Como o conjunto viável do problema (2.3) está contido no conjunto viável do problema (2.8), temos que $f(\bar{w}, \bar{b}) \leq f(w^*, b^*)$. Considere agora, para cada $t \in (0, 1)$,

$$(w_t, b_t) = (1 - t)(w^*, b^*) + t(\bar{w}, \bar{b}).$$

Para $i \in I^*$, a convexidade da restrição g_i e a viabilidade das soluções garantem que

$$g_i(w_t, b_t) \leq (1-t)g_i(w^*, b^*) + tg_i(\bar{w}, \bar{b}) \leq 0.$$

Por outro lado, para $i \notin I^*$, temos $g_i(w^*, b^*) < 0$ e portanto, pela continuidade de g_i , vale $g_i(w_t, b_t) < 0$ para t suficientemente pequeno. Portanto, (w_t, b_t) é viável para o problema (2.3).

Afirmamos que $f(\bar{w}, \bar{b}) = f(w^*, b^*)$. De fato, se $f(\bar{w}, \bar{b}) < f(w^*, b^*)$, então

$$\|w_t\| = \|(1-t)w^* + t\bar{w}\| \leq (1-t)\|w^*\| + t\|\bar{w}\| < \|w^*\|,$$

implicando $f(w_t, b_t) < f(w^*, b^*)$, o que é uma contradição. Concluimos então que (\bar{w}, \bar{b}) e (w^*, b^*) são viáveis para o problema (2.8) com o mesmo valor (ótimo) da função objetivo, ou seja, ambos são minimizadores globais do problema (2.8). Pelo Teorema 2.7, segue que $(\bar{w}, \bar{b}) = (w^*, b^*)$. \square

Conforme discutido anteriormente vamos agora apresentar o dual do problema (2.3). Para tanto considere a função Lagrangiana $L : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \rightarrow \mathbb{R}$ dada por

$$L(w, b, \alpha) = f(w, b) + \sum_{i=1}^m \alpha_i g_i(w, b) = \frac{1}{2} w^T w + \sum_{i=1}^m \alpha_i (1 - y_i (w^T x^i + b)),$$

onde f e g são dadas em (2.4) e (2.5).

Pelo que vimos no Capítulo 1, o problema dual do problema (2.3) é

$$\begin{aligned} \max_{\alpha} \quad & \inf_{w, b} L(w, b, \alpha) \\ \text{s.a.} \quad & \alpha \geq 0. \end{aligned} \tag{2.9}$$

Temos que, fixado $\alpha \in \mathbb{R}^m$, a função $(w, b) \mapsto L(w, b, \alpha)$ é convexa. De fato, f é convexa e as funções g_i , $i = 1, \dots, m$, são lineares. Então (\bar{w}, \bar{b}) minimiza globalmente $L(w, b, \alpha)$ com relação às variáveis w e b se, e somente se

$$\nabla_{(w, b)} L(\bar{w}, \bar{b}, \alpha) = 0,$$

o que equivale a

$$\begin{aligned} \frac{\partial L(\bar{w}, \bar{b}, \alpha)}{\partial w} &= \bar{w} - \sum_{i=1}^m \alpha_i y_i x^i = 0, \\ \frac{\partial L(\bar{w}, \bar{b}, \alpha)}{\partial b} &= - \sum_{i=1}^m \alpha_i y_i = 0. \end{aligned}$$

Assim, caso $\sum_{i=1}^m \alpha_i y_i = 0$, escolhendo $\bar{w} = \sum_{i=1}^m \alpha_i y_i x^i$ e $\bar{b} \in \mathbb{R}$ qualquer, temos que

$$\begin{aligned} \inf_{w,b} L(w, b, \alpha) &= L(\bar{w}, \bar{b}, \alpha) \\ &= \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j - \sum_{i=1}^m \alpha_i y_i \left(\sum_{j=1}^m \alpha_j y^j x^j \right)^T x^i - \sum_{i=1}^m \alpha_i y_i \bar{b} + \sum_{i=1}^m \alpha_i \\ &= -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i. \end{aligned}$$

Por outro lado, caso $\sum_{i=1}^m \alpha_i y_i \neq 0$, fazendo $b_k = \left(\sum_{i=1}^m \alpha_i y_i \right) k$, obtemos

$$L(0, b_k, \alpha) = \sum_{i=1}^m (\alpha_i - \alpha_i y_i b_k) = \sum_{i=1}^m \alpha_i - k \left(\sum_{i=1}^m \alpha_i y_i \right)^2,$$

donde segue que $\inf_{w,b} L(w, b, \alpha) = -\infty$. Portanto, a função dual é dada por

$$\inf_{w,b} L(w, b, \alpha) = \begin{cases} -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i, & \text{se } \sum_{i=1}^m \alpha_i y_i = 0 \\ -\infty, & \text{se } \sum_{i=1}^m \alpha_i y_i \neq 0. \end{cases}$$

Deste modo, o problema dual (2.9) pode ser reescrito como

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i \\ \text{s.a.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & \alpha_i \geq 0 \quad i = 1, \dots, m. \end{aligned} \tag{2.10}$$

Note que, os dados do conjunto de treinamento aparecem apenas como produtos internos, o que será muito importante nas próximas seções, onde trabalharemos com conjuntos não linearmente separáveis.

Veamos agora que o problema (2.10) tem solução. De fato, conforme Teorema 2.5, o problema primal (2.3) possui minimizador (w^*, b^*) . Além disso, as restrições deste problema são lineares e portanto cumprem naturalmente uma condição de qualificação [14]. Portanto, existe $\alpha^* \in \mathbb{R}_+^m$ tal que (w^*, b^*, α^*) satisfaz as condições de KKT do problema (2.3). Pelo Teorema 1.40, concluímos que α^* é maximizador global do problema dual (2.10) e que não há *gap* de dualidade.

Considere agora $\alpha^* \in \mathbb{R}_+^m$ um maximizador global arbitrário do problema dual

(2.10) e (w^*, b^*) o minimizador global do problema primal (2.3). Como o *gap* de dualidade é zero, podemos aplicar o Teorema 1.38 para concluir que (w^*, b^*, α^*) satisfaz as condições de KKT do problema (2.3). Este resultado nos fornece uma maneira de obter a solução do primal a partir de uma solução qualquer do dual.

De fato, depois de encontrado α^* , podemos obter facilmente w^* pela relação $\nabla_{(w,b)} L(w^*, b^*, \alpha^*) = 0$, ou seja,

$$w^* = \sum_{i=1}^m \alpha_i^* y_i x^i = \sum_{i \in I} \alpha_i^* y_i x^i, \quad (2.11)$$

onde $I = \{i \mid \alpha_i^* > 0\}$. Além disso, pela complementaridade, temos

$$\alpha_i^* (1 - y_i ((w^*)^T x^i + b^*)) = 0,$$

para todo $i = 1, \dots, m$. Portanto, dado $i \in I$, b^* pode ser calculado por

$$b^* = \frac{1 - y_i (w^*)^T x^i}{y_i} = y_i - \sum_{j \in I} \alpha_j^* y_j (x^j)^T x^i. \quad (2.12)$$

Note que uma solução do dual consegue detectar de modo simples quais vetores do conjunto de dados são desnecessários para a obtenção da solução primal (w^*, b^*) : todos os pontos x^i tais que $i \notin I$.

Assim, a solução primal pode ser obtida de duas formas alternativas. Uma delas, teoricamente mais simples, consiste em resolver um problema primal menor, isto é, com menos restrições, conforme o Teorema 2.8. A outra forma, computacionalmente mais simples, usa uma solução do problema dual (2.10) e as relações (2.11) e (2.12). Veremos agora que podemos simplificar ainda mais, tanto teoricamente quanto computacionalmente, conforme o que será estabelecido no Teorema 2.9.

Para tanto, se $I = \{i_1, \dots, i_p\}$, vamos denotar $\alpha_I = \begin{pmatrix} \alpha_{i_1} \\ \vdots \\ \alpha_{i_p} \end{pmatrix} \in \mathbb{R}^p$.

Teorema 2.9 *Considere (w^*, b^*) a solução do problema (2.3), $\alpha^* \in \mathbb{R}_+^m$ uma solução global arbitrária do problema dual (2.10) e $I = \{i_1, \dots, i_p\} = \{i \mid \alpha_i^* > 0\}$. Então (w^*, b^*) é a solução do problema primal reduzido*

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.a} \quad & y_i (w^T x^i + b) \geq 1, \quad i \in I \end{aligned} \quad (2.13)$$

e α_I^* é uma solução global do problema dual reduzido associado a (2.13),

$$\begin{aligned} \max_{\beta \in \mathbb{R}^p} \quad & -\frac{1}{2} \sum_{k,l=1}^p \beta_k \beta_l y_{i_k} y_{i_l} (x^{i_k})^T x^{i_l} + \sum_{k=1}^p \beta_k \\ \text{s.a.} \quad & \sum_{k=1}^p \beta_k y_{i_k} = 0, \\ & \beta_k \geq 0, \quad k = 1, \dots, p. \end{aligned} \tag{2.14}$$

Demonstração. Considere $\theta : \mathbb{R}^m \rightarrow \mathbb{R}$ a função objetivo do problema dual (2.10) e $\theta_r : \mathbb{R}^p \rightarrow \mathbb{R}$ a função objetivo do problema dual reduzido (2.14). Dado $\beta \in \mathbb{R}^p$ satisfazendo as restrições deste problema, defina o vetor $\alpha \in \mathbb{R}^m$ por

$$\alpha_i = \begin{cases} \beta_k, & \text{se } i = i_k \\ 0, & \text{se } i \notin I. \end{cases}$$

Desta forma temos $\alpha \geq 0$ e

$$\sum_{i=1}^m \alpha_i y_i = \sum_{k=1}^p \beta_k y_{i_k} = 0,$$

o que significa que α é viável para o problema (2.10). Portanto, $\theta(\alpha^*) \geq \theta(\alpha)$. Por outro lado, temos que α_I^* é viável para o problema (2.14) (pois α^* é viável para o problema (2.10)) e

$$\theta_r(\alpha_I^*) = \theta(\alpha^*) \geq \theta(\alpha) = \theta_r(\beta),$$

donde segue a segunda afirmação do teorema. Para provar a outra afirmação, note que a solução de (2.13), que vamos denotar (\bar{w}, \bar{b}) , pode ser obtida a partir de uma solução do seu dual (2.14), ou seja,

$$\bar{w} = \sum_{i \in I} \alpha_i^* y_i x^i = w^* \quad \text{e} \quad \bar{b} = y_i - \bar{w}^T x^i = b^*.$$

□

Observação: Em [25], os autores apresentam uma prova imediata da unicidade da solução do problema (2.3), recorrendo ao Teorema 1.30, ou seja, afirmando que a função objetivo $\frac{1}{2} \|w\|^2$ é estritamente convexa. Entretanto, as variáveis deste problema são w e b , e a função objetivo $(w, b) \mapsto \frac{1}{2} \|w\|^2$ não é estritamente convexa. Este argumento falho pode ser corrigido apelando ao Teorema 1.31 e observando que na solução (w^*, b^*) , temos b^* função de w^* . Esta discussão fornece uma demonstração alternativa para a unicidade da solução primal.

Mostramos que a solução do problema primal (2.3) é única. Entretanto, não podemos garantir a unicidade da solução do problema dual (2.10), conforme o Exemplo 2.10.

Exemplo 2.10 Considere o seguinte conjunto de treinamento,

$$TR = \{(0, 1), (1, 1), (2, 1), (0, -1), (1, -1), (2, -1)\},$$

com saídas $\{1, 1, 1, -1, -1, -1\}$, respectivamente, representado pela Figura 2.10. Os pontos em vermelho e azul representam as classes negativa e positiva, respectivamente.

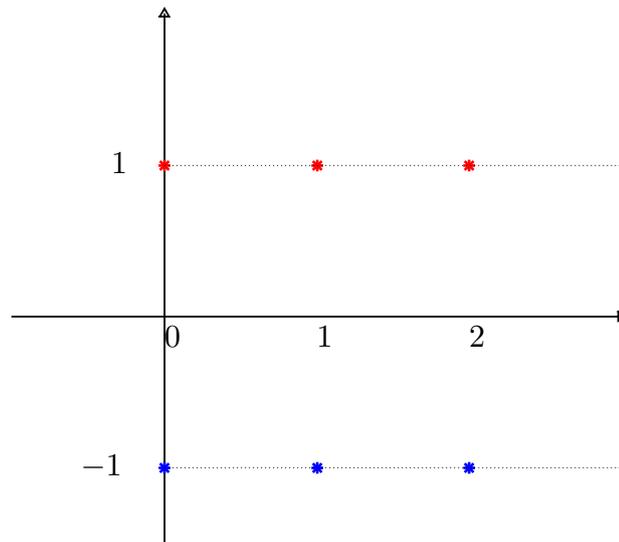


Figura 2.6: Conjunto linearmente separável.

O conjunto de treinamento é linearmente separável e resolvendo o primal (2.3), obtemos $w^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ e $b^* = 0$, fornecendo o hiperplano separador dado pela equação $x_2 = 0$.

Por outro lado, resolvendo o problema dual associado, podemos obter soluções distintas, dentre as quais, destacamos

$$\alpha^* = \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \end{bmatrix}, \quad \hat{\alpha} = \begin{bmatrix} 0 \\ 0.5 \\ 0 \\ 0.25 \\ 0 \\ 0.25 \end{bmatrix} \quad e \quad \tilde{\alpha} = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}.$$

Os vetores acima de fato são soluções pois satisfazem as condições de KKT do problema dual. Note também que usando qualquer uma destas soluções, podemos recuperar a solução primal (w^*, b^*) . Outro fato a ser observado é que o vetor α^* satisfaz a complementaridade estrita, enquanto $\hat{\alpha}$ e $\tilde{\alpha}$ não, isto é, pontos sobre os hiperplanos H_1 e H_2 estão associados a multiplicadores iguais a zero.

À luz deste exemplo vamos apresentar agora duas definições de vetores suporte, bastante presentes na literatura [1, 3, 15, 35], e fazer uma discussão a respeito delas com o objetivo de propor algumas melhorias neste conceito.

Definição 2.11 *Considere um conjunto X de vetores linearmente separáveis e (w^*, b^*) a solução do problema primal (2.3). Os vetores suporte são os vetores $x^i \in X$ tais que $y_i((w^*)^T x^i + b^*) = 1$.*

Esta definição, apesar de ser intrínseca aos dados do problema, contempla vetores que não trazem contribuição para a obtenção do hiperplano separador. De fato, conforme o Exemplo 2.10, podemos ter vetores satisfazendo $y_i((w^*)^T x^i + b^*) = 1$ com multiplicador correspondente igual a zero. Isto motiva uma segunda definição de vetores suporte.

Definição 2.12 *Considere um conjunto X de vetores linearmente separáveis e α^* uma solução do problema dual (2.10). Os vetores suporte são os vetores $x^i \in X$ tais que $\alpha_i^* > 0$.*

Ressaltamos que o conceito de vetores suporte se atribui a um conjunto de vetores e não a um determinado vetor. Assim, se considerarmos o Exemplo 2.10, os seis vetores são vetores suporte de acordo com a Definição 2.11, e também com a Definição 2.12 quando tomamos α^* . Por outro lado, se considerarmos $\hat{\alpha}$, o conjunto de vetores suporte será $\{x^2, x^4, x^6\}$.

Os vetores suporte são assim chamados pois, dentre todos os dados de treinamento, são os que possuem papel relevante, no sentido de que caso os demais pontos sejam retirados do conjunto, o classificador não muda. Isto foi provado no Teorema 2.8, para o caso da Definição 2.11 e no Teorema 2.9, para o caso da Definição 2.12. Apesar de alguns autores considerarem equivalentes estas duas definições, observe que um conjunto de vetores suporte fornecido pela Definição 2.12 é um subconjunto daqueles que cumprem a Definição 2.11. Isto significa que o cálculo de w^* , dado por (2.11), fica mais barato computacionalmente, pois são utilizados apenas os vetores que efetivamente contribuem para o cálculo do hiperplano separador.

Note ainda que, utilizando a Definição 2.12, ao representar o classificador por meio de poucos dados, o trabalho computacional para classificar futuros dados torna-se menor. Podemos classificar um novo dado x usando a definição de função de decisão 2.7, que pode ser reescrita como,

$$\begin{aligned} F(x) &= \text{sinal}((w^*)^T x + b^*) = \text{sinal}\left(\left(\sum_{i \in I} \alpha_i^* y_i x^i\right)^T x + b^*\right) \\ &= \text{sinal}\left(\sum_{i \in I} \alpha_i^* y_i (x^i)^T x + b^*\right), \end{aligned}$$

com $I = \{i \mid \alpha_i^* > 0\}$.

Um aspecto negativo, do ponto de vista teórico, da Definição 2.12 é que os vetores suporte não são definidos de modo único, pois dependem de uma solução do problema

dual. Isto pode ser observado no Exemplo 2.10. Ou seja, esta definição não é intrínseca aos dados do problema.

Também vale ressaltar que nem todo subconjunto $S \subset X \cap (H1 \cup H2)$ cuja solução do problema primal reduzido (isto é, considerando os dados $x^i \in S$) seja (w^*, b^*) , será um conjunto de vetores suporte de acordo com esta última definição. De fato, considere os vetores $(0, 1)$, $(1, 1)$ e $(1, -1)$ do Exemplo 2.10. Se existisse uma solução do problema dual cujas componentes correspondentes a estes vetores fossem não nulas e as outras componentes nulas, o Teorema 2.9 garantiria que o dual reduzido teria uma solução com as três componentes não nulas. Mas isto é impossível pois o dual reduzido tem uma única solução na qual a primeira componente é nula.

Nossa proposta é utilizar o Teorema 2.9 recursivamente e resolver uma sequência de problemas duais, com dimensões cada vez menores, para no final obtermos uma solução dual com o maior número de componentes nulas possível. Consequentemente, obteremos uma quantidade reduzida de vetores suporte, tornando o cálculo de w^* , bem como do classificador de novos pontos, mais barato computacionalmente.

Algoritmo 2.1

Dado: Um conjunto de treinamento linearmente separável $X = \{(x^1, y_1), \dots, (x^m, y_m)\}$.

Resolva o problema dual (2.10), obtendo uma solução α^* .

Calcule $I = \{i \mid \alpha_i^* > 0\}$ e $X = \{(x^i, y_i) \in X \mid i \in I\}$.

REPITA

Resolva o problema dual reduzido (2.14), obtendo uma solução α^* .

Calcule $I = \{i \mid \alpha_i^* > 0\}$ e $X = \{(x^i, y_i) \in X \mid i \in I\}$.

FIM

Saída: $\alpha_I^* \in \mathbb{R}^p$, $p \ll m$ e $X = \{(x^i, y_i) \in X \mid i \in I\}$.

Em relação ao critério de parada do algoritmo, não sabemos o que seria mais barato computacionalmente: procurar uma solução dual mais esparsa e utilizá-la para a obtenção do classificador e para a classificação de dados futuros ou utilizar a solução dual obtida, mesmo que sem uma quantidade considerável de componentes iguais a zero.

Nesta seção, discutimos SVM apenas para dados linearmente separáveis, obtendo a máxima margem entre eles. Entretanto, são raros os problemas reais onde os dados sejam linearmente separáveis. Nas próximas seções, trataremos os problemas onde isso não ocorre.

2.3 Máquinas de Vetores Suporte - CSVM

Estenderemos agora para dados não linearmente separáveis os conceitos vistos na Seção 2.2. Considere então um conjunto de dados onde a hipótese de ser linearmente separável não seja satisfeita, como por exemplo, o da Figura 2.7.

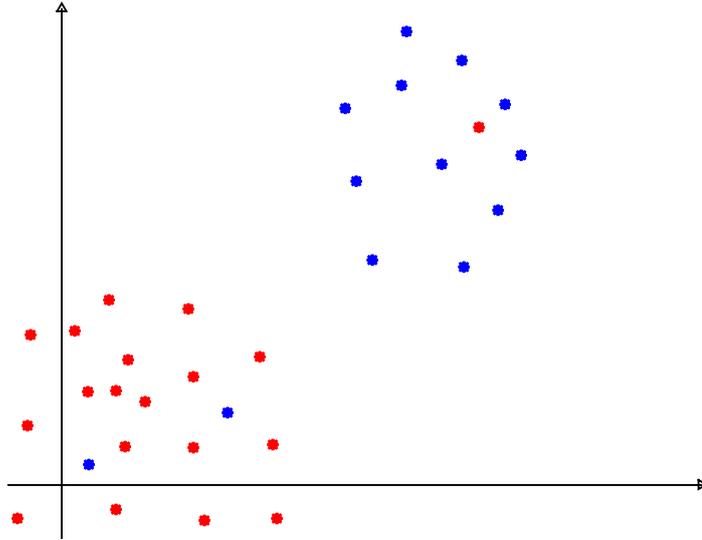


Figura 2.7: Conjunto não linearmente separável.

Neste caso, não existe um hiperplano separador, ou seja, a formulação dada pelo problema (2.3) não fornece um classificador, já que o seu conjunto viável

$$\{(w, b) \in \mathbb{R}^{n+1} \mid 1 - y_i(w^T x^i + b) \leq 0, i = 1, \dots, m\}$$

é vazio.

Para contornar essa situação, podemos utilizar regularização, permitindo a violação das restrições do problema (2.3). Isto é feito acrescentando variáveis de folga $\xi_i \geq 0$, associadas aos dados de treinamento $x^i, i = 1, \dots, m$, flexibilizando assim o problema de estimar as variáveis w e b . Mais precisamente, a restrição $1 - y_i(w^T x^i + b) \leq 0$ é relaxada e substituída por $1 - y_i(w^T x^i + b) \leq \xi_i$, com $\xi_i \geq 0$, o que equivale a

$$w^T x^i + b \geq 1 - \xi_i, \text{ para todo } x^i \in X_1 \quad (2.15)$$

$$w^T x^j + b \leq -1 + \xi_j, \text{ para todo } x^j \in X_2. \quad (2.16)$$

Assim, é possível aceitar dados não pertencentes ao semiespaço correspondente. Isto é, podemos ter pontos da classe positiva fora do semiespaço $S_1 = \{x \in \mathbb{R}^n \mid w^T x + b \geq 1\}$ e/ou pontos da classe negativa fora do semiespaço $S_2 = \{x \in \mathbb{R}^n \mid w^T x + b \leq -1\}$. Neste caso, o hiperplano separador é denominado hiperplano de margem flexível [21].

Para ilustrar as variáveis de folga, destacamos na Figura 2.8 três pontos da classe positiva, x^i, x^j e x^ℓ , para os quais vemos que

$$w^T x^i + b > 1, \quad 0 < w^T x^j + b < 1 \quad \text{e} \quad w^T x^\ell + b < 0.$$

Portanto,

$$w^T x^i + b \geq 1 - \xi_i, \quad w^T x^j + b \geq 1 - \xi_j \quad \text{e} \quad w^T x^\ell + b \geq 1 - \xi_\ell$$

para $\xi_i = 0$, $0 < \xi_j < 1$ e $\xi_\ell > 1$.

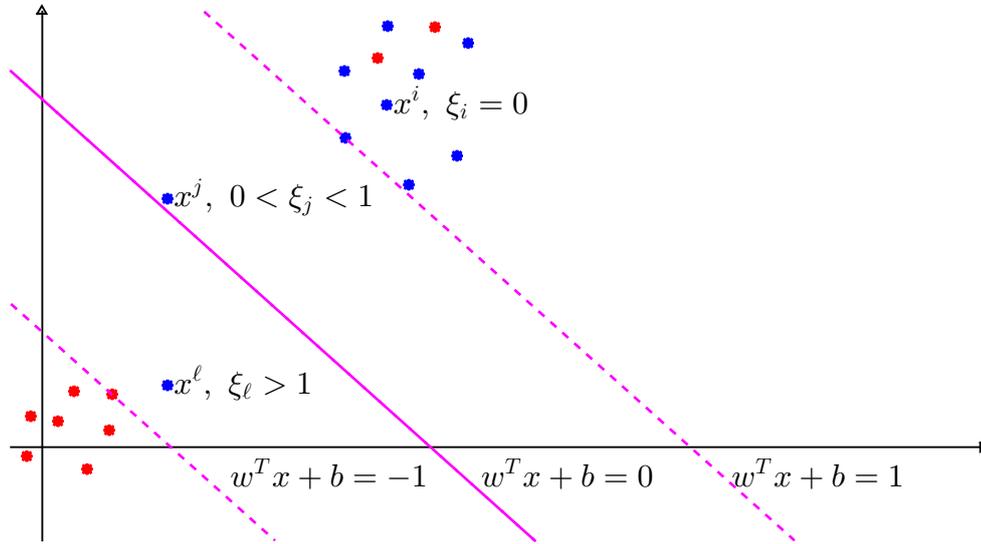


Figura 2.8: Separação de um conjunto de dados usando variáveis de folga.

A questão agora é como obter w e b ótimos, para termos um bom classificador. Note que dados w e b arbitrários, podemos escolher $\xi_i \geq 0$ de modo que as restrições (2.15) e (2.16) sejam satisfeitas. De fato, basta definir

$$\xi_i = \begin{cases} \max\{0, 1 - w^T x^i - b\}, & \text{se } x^i \in X_1 \\ \max\{0, 1 + w^T x^i + b\}, & \text{se } x^i \in X_2. \end{cases}$$

Desta forma, para obter um bom classificador, não basta apenas pensar em maximizar a margem definida pelos hiperplanos $H_1 = \{x \in \mathbb{R}^n \mid w^T x + b = 1\}$ e $H_2 = \{x \in \mathbb{R}^n \mid w^T x + b = -1\}$. Na situação ilustrada na Figura 2.9 vemos claramente que o hiperplano dado por $w_0^T x + b_0 = 0$ não pode ser usado para classificar os dados. Este mesmo hiperplano pode ser descrito por $w^T x + b = 0$, com $w = \frac{w_0}{k}$ e $b = \frac{b_0}{k}$, qualquer que seja $k \in \mathbb{N}$. Além disso, a margem $\frac{2}{\|w\|} = \frac{2k}{\|w_0\|}$ pode ficar arbitrariamente grande. Ou seja, não adianta introduzir as variáveis de folga apenas nas restrições e manter a mesma função objetivo.

Deste modo, para reformular o problema original de maximizar a margem, acrescentando as variáveis de folga, precisamos também controlar o valor destas variáveis de modo a reduzir o número de pontos classificados incorretamente. Quanto maior o valor delas, mais será permitido violar as restrições. Assim, colocamos na função objetivo uma

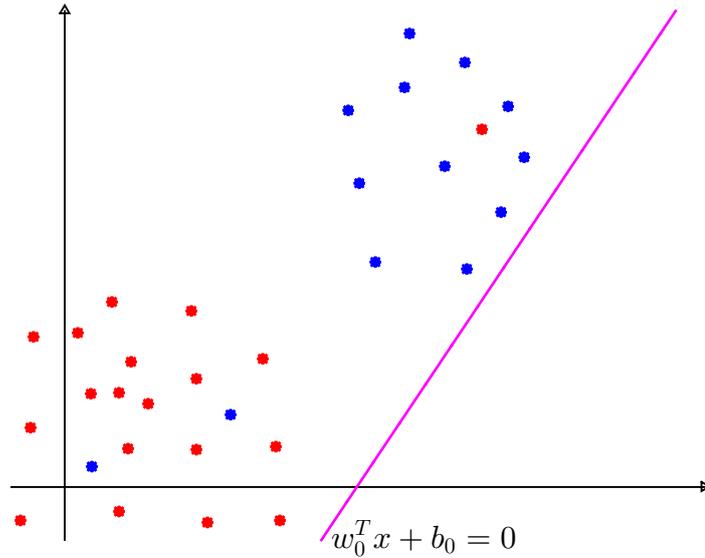


Figura 2.9: Vetor (w_0, b_0, ξ) satisfazendo (2.15) e (2.16).

parcela que corresponde à penalização das violações. Isto pode ser feito considerando o problema

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s.a.} \quad & y_i(w^T x^i + b) \geq 1 - \xi_i, \quad i = 1, \dots, m, \\ & \xi_i \geq 0, \quad i = 1, \dots, m \end{aligned} \quad (2.17)$$

onde $C > 0$ é um parâmetro de regularização que serve para controlar a importância das variáveis de folga. Pela utilização deste parâmetro, esta modelagem de SVM também é conhecido como C-SVM.

O melhor valor para o parâmetro C , que forneceria uma boa classificação dos dados, não é conhecido a priori. Na fase de treinamento, escolhamos o valor de C de forma heurística, geralmente a partir da natureza do problema.

O termo $C \sum_{i=1}^m \xi_i$ na função objetivo do problema (2.17) minimiza o valor das variáveis de folga, reduzindo assim o número de pontos classificados incorretamente, podendo ser pensado também como uma medida de erro de classificação. De fato, aumentando o valor do parâmetro C , aumenta-se a penalização sobre a violação da restrição original do problema SVM. Por outro lado, diminuindo o valor de C , o modelo se torna mais flexível a esse tipo de violação.

É importante ressaltar que apesar das variáveis de folga serem introduzidas tanto nas restrições quanto na função objetivo do problema primal, a função de decisão, usada para classificar novos dados, é a mesma do caso de margem rígida.

Note que o problema de margem flexível (2.17), assim como o problema (2.3), também possui restrições lineares $g_i(w, b, \xi) = 1 - \xi_i - y_i(w^T x^i + b) \leq 0$ e $h_i(w, b, \xi) =$

$-\xi_i \leq 0, i = 1, \dots, m$ e assim o conjunto viável

$$\Omega = \{(w, b, \xi) \in \mathbb{R}^{n+1+m} \mid g_i(w, b, \xi) \leq 0, h_i(w, b, \xi) \leq 0, i = 1, \dots, m\}$$

é um poliedro não vazio. Além disso, a função objetivo f é quadrática e limitada inferiormente em Ω , pois

$$f(w, b, \xi) = \frac{1}{2}\|w\|^2 + \underbrace{C}_{>0} \sum_{i=1}^m \underbrace{\xi_i}_{\geq 0} \geq 0.$$

Assim, o Teorema 1.34 garante a existência de um minimizador global para o problema (2.17). De qualquer modo, apresentamos aqui uma demonstração alternativa da existência de um minimizador global para este problema.

Teorema 2.13 *O problema (2.17) possui um minimizador global.*

Demonstração. Como a função objetivo do problema (2.17) é limitada inferiormente e o conjunto viável é não vazio, temos que

$$f^* \stackrel{\text{def}}{=} \inf_{(w,b,\xi) \in \Omega} f(w, b, \xi) > -\infty.$$

Pela definição de ínfimo, podemos concluir que para todo $k \in \mathbb{N}$, existe $(w_k, b_k, \xi_k) \in \Omega$ tal que

$$f^* \leq f(w_k, b_k, \xi_k) < f^* + \frac{1}{k}.$$

Deste modo,

$$\frac{1}{2}\|w_k\|^2 + C \sum_{i=1}^m (\xi_k)_i = f(w_k, b_k, \xi_k) \rightarrow f^* \quad (2.18)$$

Como

$$0 \leq \frac{1}{2}\|w_k\|^2 \leq f(w_k, b_k, \xi_k) \quad \text{e} \quad 0 \leq C \sum_{i=1}^m (\xi_k)_i \leq f(w_k, b_k, \xi_k),$$

temos que as sequências (w_k) e (ξ_k) são limitadas. Portanto, (b_k) também é limitada. De fato, dados $x^j \in X_1$ e $x^\ell \in X_2$ arbitrários, por (2.15) e (2.16), obtemos

$$1 - (\xi_k)_j - w_k^T x^j \leq b_k \leq -1 + (\xi_k)_\ell - w_k^T x^\ell.$$

Como a sequência (w_k, b_k, ξ_k) é limitada, pelo Teorema 1.3, possui uma subsequência convergente, seja ela $(w_k, b_k, \xi_k) \xrightarrow{\mathbb{N}'} (w^*, b^*, \xi^*)$. Logo, pela continuidade de f obtemos

$$f(w_k, b_k, \xi_k) \xrightarrow{\mathbb{N}'} f(w^*, b^*, \xi^*),$$

e por (2.18) segue que $f(w^*, b^*, \xi^*) = f^*$. Portanto, provamos que existe $(w^*, b^*, \xi^*) \in \Omega$ tal que

$$f(w^*, b^*, \xi^*) = f^* \leq f(w, b, \xi)$$

para todo $(w, b, \xi) \in \Omega$. □

Diferentemente do caso em que os dados são linearmente separáveis, para o problema (2.17), somente sob certas condições podemos garantir a unicidade da solução. Veremos adiante, no Exemplo 2.16, que podemos ter infinitas soluções. Entretanto, conforme o resultado que segue, a componente w de qualquer solução é única. Este resultado pode ser encontrado em [25], mas por conveniência apresentamos aqui com algumas modificações no enunciado e na demonstração.

Teorema 2.14 *Sejam (w^*, b^*, ξ^*) e $(\bar{w}, \bar{b}, \bar{\xi})$ soluções arbitrárias do problema (2.17). Então $w^* = \bar{w}$.*

Demonstração. Como o problema (2.17) é convexo, temos

$$f(w^*, b^*, \xi^*) = f(\bar{w}, \bar{b}, \bar{\xi}) = f^* \stackrel{\text{def}}{=} \inf_{(w, b, \xi) \in \Omega} f(w, b, \xi).$$

Mais ainda, definindo $(w_t, b_t, \xi_t) = (1 - t)(w^*, b^*, \xi^*) + t(\bar{w}, \bar{b}, \bar{\xi})$, temos que

$$\frac{1}{2} \|w_t\|^2 + C \sum_{i=1}^m (\xi_t)_i = f(w_t, b_t, \xi_t) = f^*$$

para todo $t \in [0, 1]$. Portanto,

$$\frac{1}{2} \sum_{i=1}^n ((1 - t)w_i^* + t\bar{w}_i)^2 + C \sum_{i=1}^m ((1 - t)\xi_i^* + t\bar{\xi}_i) = f^*$$

para todo $t \in [0, 1]$. Derivando em relação a t , obtemos

$$\sum_{i=1}^n ((1 - t)w_i^* + t\bar{w}_i)(\bar{w}_i - w_i^*) + C \sum_{i=1}^m (\bar{\xi}_i - \xi_i^*) = 0.$$

Derivando novamente em relação a t , obtemos

$$\sum_{i=1}^n (\bar{w}_i - w_i^*)^2 = 0,$$

donde segue que $w^* = \bar{w}$. □

No que segue precisaremos das condições de KKT para o problema primal (2.17). Como as restrições deste problema são lineares, elas satisfazem uma condição de qualificação. Portanto, dada qualquer solução (w^*, b^*, ξ^*) , existem multiplicadores $\alpha^*, \beta^* \in \mathbb{R}^m$

tais que $(w^*, b^*, \xi^*, \alpha^*, \beta^*)$ satisfaz

$$\begin{pmatrix} w^* \\ 0 \\ Ce \end{pmatrix} - \begin{pmatrix} \sum_{i=1}^m \alpha_i^* y_i x^i \\ \sum_{i=1}^m \alpha_i^* y_i \\ \alpha^* + \beta^* \end{pmatrix} = 0 \quad (2.19)$$

$$\alpha_i^* (1 - \xi_i^* - y_i((w^*)^T x^i + b^*)) = 0, \quad i = 1, \dots, m, \quad (2.20)$$

$$\beta_i^* \xi_i^* = 0, \quad i = 1, \dots, m, \quad (2.21)$$

$$1 - \xi_i^* - y_i((w^*)^T x^i + b^*) \leq 0, \quad i = 1, \dots, m, \quad (2.22)$$

$$\alpha_i^* \geq 0, \quad \beta_i^* \geq 0, \quad \xi_i^* \geq 0, \quad i = 1, \dots, m, \quad (2.23)$$

onde e representa o vetor com todas as componentes iguais a 1.

É interessante notar que um ponto x^i está no semiespaço correto definido pela margem se, e somente se, a sua variável de folga vale zero. Além disso, caso isto seja violado, o valor da variável de folga é exatamente a medida da violação. Isto é formalizado no seguinte resultado.

Teorema 2.15 *Temos $y_i((w^*)^T x^i + b^*) \geq 1$ se, e somente se, $\xi_i^* = 0$. Além disso, se $\xi_i^* > 0$, então $\xi_i^* = 1 - y_i((w^*)^T x^i + b^*)$.*

Demonstração. Caso $y_i((w^*)^T x^i + b^*) > 1$, por (2.20) temos que $\alpha_i^* = 0$. Assim, segue de (2.19) que $\beta_i^* = C$ e portanto $\xi_i^* = 0$ por (2.21). Caso $y_i((w^*)^T x^i + b^*) = 1$, por (2.19), (2.20) e (2.21) temos que $C\xi_i^* = (\alpha_i^* + \beta_i^*)\xi_i^* = 0$, donde segue que $\xi_i^* = 0$. Reciprocamente, se $\xi_i^* = 0$, então $y_i((w^*)^T x^i + b^*) \geq 1$ por (2.22). Por fim, se $\xi_i^* > 0$, então $\beta_i^* = 0$ e portanto, $\alpha_i^* = C$, o que implica $\xi_i^* = 1 - y_i((w^*)^T x^i + b^*)$. \square

Pelos Teoremas 2.14 e 2.15, o vetor ξ^* fica completamente determinado por w^* e b^* , uma vez que

$$\xi_i^* = \max\{0, 1 - y_i((w^*)^T x^i + b^*)\}.$$

Assim, se existir mais do que uma solução, obrigatoriamente a componente b^* deve ser diferente.

Exemplo 2.16 *Considere o conjunto de dados $X = \{-1, 1\}$, com saídas $Y = \{-1, 1\}$, e o problema primal associado*

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2}w^2 + C(\xi_1 + \xi_2) \\ \text{s.a} \quad & 1 - \xi_1 - w + b \leq 0 \end{aligned} \quad (2.24)$$

$$1 - \xi_2 - w - b \leq 0 \quad (2.25)$$

$$-\xi_1 \leq 0 \quad (2.26)$$

$$-\xi_2 \leq 0, \quad (2.27)$$

onde $C > 0$. Então, para $C \geq \frac{1}{2}$ o problema tem uma única solução e caso $C < \frac{1}{2}$ temos uma infinidade de soluções. De fato, pelas condições de KKT para tal problema, temos que

$$\begin{pmatrix} w \\ 0 \\ C \\ C \end{pmatrix} + \alpha_1 \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} -1 \\ -1 \\ 0 \\ -1 \end{pmatrix} + \beta_1 \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} + \beta_2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.28)$$

$$\alpha_1(1 - \xi_1 - w + b) = 0 \quad (2.29)$$

$$\alpha_2(1 - \xi_2 - w - b) = 0 \quad (2.30)$$

$$\beta_1 \xi_1 = 0 \quad (2.31)$$

$$\beta_2 \xi_2 = 0. \quad (2.32)$$

Da condição (2.28), obtemos

$$\begin{pmatrix} w - \alpha_1 - \alpha_2 \\ \alpha_1 - \alpha_2 \\ C - \alpha_1 - \beta_1 \\ C - \alpha_2 - \beta_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

o que equivale a

$$\alpha_1 = \alpha_2 \quad (2.33)$$

$$w = \alpha_1 + \alpha_2 = 2\alpha_1 \quad (2.34)$$

$$C = \alpha_1 + \beta_1 = \alpha_2 + \beta_2, \quad (2.35)$$

Afirmamos que $\alpha_1 \neq 0$. Suponha por absurdo que $\alpha_1 = 0$. Então por (2.33)-(2.35) obtemos $\alpha_2 = 0$, $w = 0$ e $\beta_1 = \beta_2 = C > 0$. Das condições (2.31) e (2.32) segue que $\xi_1 = \xi_2 = 0$ e assim, pelas restrições (2.24) e (2.25), temos $b \leq -1$ e $b \geq 1$, o que é uma contradição. Deste modo, por (2.33), (2.29), (2.30), (2.26) e (2.27) obtemos

$$\begin{aligned} 1 - w + b &= \xi_1 \geq 0 \\ 1 - w - b &= \xi_2 \geq 0, \end{aligned} \quad (2.36)$$

logo, $-1 + w \leq b \leq 1 - w$, o que implica que $w \leq 1$. Somando (2.31) com (2.32) e também subtraindo, usando o fato de que $\beta_2 = \beta_1$ e (2.36) obtemos

$$\begin{aligned} \beta_1(\xi_1 + \xi_2) &= 0, \text{ o que implica } \beta_1(1 - w) = 0, \\ \beta_1(\xi_1 - \xi_2) &= 0, \text{ o que implica } \beta_1 b = 0. \end{aligned} \quad (2.37)$$

Se $\beta_1 = 0$ então por (2.35) temos $C = \alpha_1$ e conseqüentemente por (2.34) obtemos $2C = w$,

mas $w \leq 1$, logo $C \leq \frac{1}{2}$.

Portanto, se $C > \frac{1}{2}$ então $\beta_1 \neq 0$, o que implica $\xi_1 = \xi_2 = 0$ por (2.31) e (2.32), já por (2.37) temos $w = 1$ e $b = 0$, fornecendo assim uma única solução.

Considere agora $C = \frac{1}{2}$. Afirmamos que $w = 1$. De fato, se $\beta_1 = 0$ por (2.35) e (2.34) temos $\frac{1}{2} = C = \alpha_1 = \frac{w}{2}$. Agora, se $\beta_1 \neq 0$ por (2.37) obtemos $w = 1$. Portanto, por (2.36) temos $b = 0$ e $\xi_1 = \xi_2 = 0$. Neste caso também temos uma única solução.

Por fim, considere $C < \frac{1}{2}$. Defina $w = 2C$, $-1 + 2C \leq b \leq 1 - 2C$, $\xi_1 = 1 - 2C + b$, $\xi_2 = 1 - 2C - b$, $\alpha_1 = \alpha_2 = C$ e $\beta_1 = \beta_2 = 0$. Desta forma, as condições de KKT são satisfeitas. Como $-1 + 2C < 1 - 2C$, temos neste caso uma infinidade de soluções.

Intuitivamente, se o conjunto de dados fosse linearmente separável, ao resolver o problema (2.17) teríamos uma solução única e igual à obtida pelo problema (2.3). Entretanto, conforme o exemplo acima, isto não procede. Além disso, baseado em [35], temos que quanto maior o valor de C , menos erros de classificação são permitidos. De fato, no exemplo acima, para $C \geq \frac{1}{2}$ o hiperplano obtido separa corretamente os dados. Se $C < \frac{1}{2}$ teremos soluções que separam corretamente os dados e outras que não.

Vamos agora apresentar os resultados tirados de [25] a respeito da unicidade da solução para o problema (2.17). Novamente por conveniência e também para esclarecer as ideias, reestruturamos os enunciados e as demonstrações.

Teorema 2.17 *Considere um conjunto de dados $X \subset \mathbb{R}^n$, com saída $y \in \mathbb{R}^n$, uma constante $C > 0$ e o problema primal (2.17) associado. Considere também uma solução (w^*, b^*, ξ^*) deste problema e os conjuntos de índices*

$$\begin{aligned} N_1 &= \{i \mid y_i = 1 \text{ e } (w^*)^T x^i + b^* < 1\}, \\ N_2 &= \{i \mid y_i = -1 \text{ e } (w^*)^T x^i + b^* > -1\}, \\ N_3 &= \{i \mid y_i = 1 \text{ e } (w^*)^T x^i + b^* = 1\}, \\ N_4 &= \{i \mid y_i = -1 \text{ e } (w^*)^T x^i + b^* = -1\}, \\ N_5 &= \{i \mid y_i = 1 \text{ e } (w^*)^T x^i + b^* > 1\} \text{ e} \\ N_6 &= \{i \mid y_i = -1 \text{ e } (w^*)^T x^i + b^* < -1\}. \end{aligned}$$

Então a solução é única se, e somente se,

$$|N_2 \cup N_4| \neq |N_1| \quad \text{e} \quad |N_1 \cup N_3| \neq |N_2|. \quad (2.38)$$

Demonstração. Para provar a necessidade, suponha por absurdo que a condição (2.38) não ocorra. Então temos duas possibilidades: $|N_2 \cup N_4| = |N_1|$ ou $|N_1 \cup N_3| = |N_2|$. Considere inicialmente a primeira delas e defina $\delta = \min \left\{ \min_{i \in N_1} \xi_i^*, \min_{i \in N_6} (-1 - (w^*)^T x^i - b^*) \right\}$, $w' = w^*$, $b' = b^* + \delta$, $\xi'_i = \xi_i^* - \delta$, para $i \in N_1$, $\xi'_i = \xi_i^* + \delta$, para $i \in N_2 \cup N_4$ e $\xi'_i = 0$ para os outros casos. Chegaremos em uma contradição provando que (w', b', ξ') também é uma

solução do problema e é diferente de (w^*, b^*, ξ^*) . Para isto, note primeiro que se ocorresse $N_1 \cup N_6 = \emptyset$, teríamos $N_2 \cup N_4 = \emptyset$ e assim não existiriam pontos da classe negativa. Portanto, δ está bem definido. Além disso, pelo Teorema 2.15 e definição de N_6 temos que $\delta > 0$, provando que os vetores (w', b', ξ') e (w^*, b^*, ξ^*) são de fato diferentes. Temos também que

$$\sum_{i=1}^m \xi'_i = \sum_{i \in N_1} (\xi_i^* - \delta) + \sum_{i \in N_2 \cup N_4} (\xi_i^* + \delta) = \sum_{i \in N_1 \cup N_2 \cup N_4} \xi_i^* = \sum_{i=1}^m \xi_i^*,$$

onde a última igualdade decorre do Teorema 2.15. Portanto,

$$f(w', b', \xi') = \frac{1}{2} \|w'\|^2 + C \sum_{i=1}^m \xi'_i = \frac{1}{2} \|w^*\|^2 + C \sum_{i=1}^m \xi_i^* = f(w^*, b^*, \xi^*).$$

Além disso, as restrições são satisfeitas em (w', b', ξ') . De fato, se $i \in N_1$ e por $w^* = w'$ pelo Teorema 2.14 temos

$$y_i(w'^T x^i + b') = (w^*)^T x^i + b^* + \delta \geq 1 - \xi_i^* + \delta = 1 - \xi'_i.$$

Para $i \in N_2 \cup N_4$, temos

$$y_i(w'^T x^i + b') = y_i((w^*)^T x^i + b^*) - \delta \geq 1 - \xi_i^* - \delta = 1 - \xi'_i.$$

Caso $i \in N_3 \cup N_5$, temos

$$y_i(w'^T x^i + b') = (w^*)^T x^i + b^* + \delta \geq 1 + \delta \geq 1 - \xi'_i.$$

Se $i \in N_6$, então $\xi'_i = 0$ e $\delta \leq -1 - (w^*)^T x^i - b^*$. Assim,

$$y_i(w'^T x^i + b') = -(w^*)^T x^i - b^* - \delta \geq 1 = 1 - \xi'_i.$$

Finalmente, note que pela definição de δ , temos $\xi'_i \geq 0$ para todo $i = 1, \dots, m$. Desta forma, caso $|N_2 \cup N_4| = |N_1|$, encontramos uma solução diferente de (w^*, b^*, ξ^*) , contradizendo a hipótese. Resta agora analisar o caso em que $|N_1 \cup N_3| = |N_2|$. Neste caso, definindo $\delta = \min \left\{ \min_{i \in N_2} \xi_i^*, \min_{i \in N_5} ((w^*)^T x^i + b^* - 1) \right\}$, $w' = w^*$, $b' = b^* - \delta$, $\xi'_i = \xi_i^* - \delta$, para $i \in N_2$, $\xi'_i = \xi_i^* + \delta$, para $i \in N_1 \cup N_3$ e $\xi'_i = 0$ para os outros casos, podemos provar, de modo análogo ao caso anterior, que (w', b', ξ') é uma solução diferente de (w^*, b^*, ξ^*) , novamente contradizendo a hipótese.

Vejam agora que a condição (2.38) é suficiente. Suponha por absurdo que exista outra solução (w', b', ξ') diferente de (w^*, b^*, ξ^*) . Pelo Teorema 2.14, temos $w' = w^*$. Portanto, em virtude do Teorema 2.15, $b' \neq b^*$, já que a variável de folga fica completamente determinada a partir de w e b . Além disso, como o conjunto solução é

convexo, para qualquer $t \in [0, 1]$, definindo $\delta = t(b' - b^*)$ e $\tilde{\xi} = (1 - t)\xi^* + t\xi'$, temos que

$$(1 - t)(w^*, b^*, \xi^*) + t(w', b', \xi') = (w^*, b^* + \delta, \tilde{\xi})$$

também será uma solução. Temos dois casos para analisar: $\delta > 0$ e $\delta < 0$. Suponha primeiro que $\delta > 0$. Podemos escolher $t \in (0, 1]$ suficientemente pequeno de modo que $(w^*)^T x^i + b^* + \delta < 1$ para todo $i \in N_1$ e $(w^*)^T x^i + b^* + \delta < -1$ para todo $i \in N_6$. Assim,

$$\tilde{\xi}_i = 1 - (w^*)^T x^i - b^* - \delta = \xi_i^* - \delta, \quad i \in N_1 \quad \text{e} \quad \tilde{\xi}_i = 0, \quad i \in N_6. \quad (2.39)$$

Para $i \in N_2 \cup N_4$, o Teorema 2.15 nos permite concluir que

$$y_i((w^*)^T x^i + b^* + \delta) = y_i((w^*)^T x^i + b^*) - \delta = 1 - (\xi_i^* + \delta) < 1.$$

Portanto, este mesmo teorema implica

$$\tilde{\xi}_i = 1 - y_i((w^*)^T x^i + b^* + \delta) = \xi_i^* + \delta, \quad i \in N_2 \cup N_4. \quad (2.40)$$

Finalmente, caso $i \in N_3 \cup N_5$, temos

$$y_i((w^*)^T x^i + b^* + \delta) = (w^*)^T x^i + b^* + \delta \geq 1 + \delta \geq 1,$$

donde segue que

$$\tilde{\xi}_i = \xi_i^* = 0, \quad i \in N_3 \cup N_5. \quad (2.41)$$

Como $(w^*, b^* + \delta, \tilde{\xi})$ é solução do problema, temos

$$\frac{1}{2}\|w^*\|^2 + C \sum_{i=1}^m \tilde{\xi}_i = f(w^*, b^* + \delta, \tilde{\xi}) = f(w^*, b^*, \xi^*) = \frac{1}{2}\|w^*\|^2 + C \sum_{i=1}^m \xi_i^*.$$

Mas isto, junto com (2.39), (2.40) e (2.41), implica

$$\sum_{i \in N_1} (\xi_i^* - \delta) + \sum_{i \in N_2 \cup N_4} (\xi_i^* + \delta) = \sum_{i=1}^m \tilde{\xi}_i = \sum_{i=1}^m \xi_i^* = \sum_{i \in N_1 \cup N_2 \cup N_4} \xi_i^*.$$

Desta forma, obtemos

$$-\delta|N_1| + \delta|N_2 \cup N_4| = 0,$$

contradizendo (2.38). Finalmente, para o caso $\delta < 0$ pode se provar de modo análogo que $|N_2| = |N_1 \cup N_3|$. \square

Em analogia com o que foi feito para o caso linearmente separável, aqui também será conveniente considerar a formulação dual do problema (2.17) com o propósito de obter um classificador. Para isto, considere

a função Lagrangiana $L : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ dada por

$$\begin{aligned} L(w, b, \xi, \alpha, \beta) &= f(w, b, \xi) + \sum_{i=1}^m \alpha_i g_i(w, b, \xi) + \sum_{j=1}^m \beta_j h_j(w, b, \xi) \\ &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i + \sum_{i=1}^m \alpha_i (1 - \xi_i - y_i (w^T x^i + b)) - \sum_{i=1}^m \beta_i \xi_i \end{aligned}$$

com $\alpha_i \geq 0$ o multiplicador de Lagrange associado a $g_i(w, b, \xi)$ e $\beta_i \geq 0$ o multiplicador de Lagrange associado a $h_i(w, b, \xi)$, $i = 1, \dots, m$. É importante ressaltar que h_i , $i = 1, \dots, m$ são também referentes a restrições de desigualdade.

Pelo que vimos no Capítulo 1, o problema dual de (2.17) é dado por

$$\begin{aligned} \max_{\alpha, \beta} \quad & \inf_{w, b, \xi} L(w, b, \xi, \alpha, \beta) \\ \text{s.a} \quad & \alpha \geq 0, \\ & \beta \geq 0. \end{aligned} \tag{2.42}$$

Note que, fixados $\alpha, \beta \in \mathbb{R}^m$, a função $(w, b, \xi) \mapsto L(w, b, \xi, \alpha, \beta)$ é convexa, pois f é convexa e as funções g_i, h_i , $i = 1, \dots, m$ são lineares. Então $(\bar{w}, \bar{b}, \bar{\xi})$ minimiza globalmente $L(w, b, \xi, \alpha, \beta)$ com relação às variáveis w, b e ξ se, e somente se

$$\nabla_{(w, b, \xi)} L(\bar{w}, \bar{b}, \bar{\xi}, \alpha, \beta) = 0,$$

o que equivale a

$$\begin{aligned} \frac{\partial L(\bar{w}, \bar{b}, \bar{\xi}, \alpha, \beta)}{\partial w} &= \bar{w} - \sum_{i=1}^m \alpha_i y_i x^i = 0, \\ \frac{\partial L(\bar{w}, \bar{b}, \bar{\xi}, \alpha, \beta)}{\partial b} &= - \sum_{i=1}^m \alpha_i y_i = 0, \\ \frac{\partial L(\bar{w}, \bar{b}, \bar{\xi}, \alpha, \beta)}{\partial \xi} &= Ce - \alpha - \beta = 0. \end{aligned}$$

onde e representa o vetor com todas as componentes iguais a 1.

Assim, caso $\sum_{i=1}^m \alpha_i y_i = 0$ e $\alpha + \beta = Ce$, escolhendo $\bar{w} = \sum_{i=1}^m \alpha_i y_i x^i$, $\bar{b} \in \mathbb{R}$ e

$\bar{\xi} \in \mathbb{R}^m$ quaisquer, temos que

$$\begin{aligned}
\inf_{w,b,\xi} L(w, b, \xi, \alpha, \beta) &= L(\bar{w}, \bar{b}, \bar{\xi}, \alpha, \beta) \\
&= \frac{1}{2} \left\| \sum_{i=1}^m \alpha_i y_i x^i \right\|^2 + \sum_{i=1}^m (C - \beta_i) \bar{\xi}_i \\
&\quad - \sum_{i=1}^m \alpha_i (y_i (\bar{w}^T x^i + \bar{b}) - 1 + \bar{\xi}_i) \\
&= \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \underbrace{(C - \beta_i - \alpha_i)}_{=0} \bar{\xi}_i \\
&\quad - \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j - \sum_{i=1}^m \alpha_i y_i \bar{b} + \sum_{i=1}^m \alpha_i \\
&= -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i - \bar{b} \underbrace{\sum_{i=1}^m \alpha_i y_i}_{=0} \\
&= -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i.
\end{aligned}$$

Por outro lado, caso $\sum_{i=1}^m \alpha_i y_i \neq 0$, fazendo $b_k = k \sum_{i=1}^m \alpha_i y_i$, obtemos

$$L(0, b_k, 0, \alpha, \beta) = - \sum_{i=1}^m (\alpha_i y_i b_k - \alpha_i) = -k \left(\sum_{i=1}^m \alpha_i y_i \right)^2 + \sum_{i=1}^m \alpha_i,$$

donde segue que $\inf_{w,b,\xi} L(w, b, \xi, \alpha, \beta) = -\infty$. Finalmente, caso $\alpha_j + \beta_j \neq C$ para algum $j \in \{1, \dots, m\}$, defina ξ_k por $(\xi_k)_j = -k(C - \alpha_j - \beta_j)$ e $(\xi_k)_i = 0$ para $i \neq j$. Assim,

$$L(0, 0, \xi_k, \alpha, \beta) = -k(C - \alpha_j - \beta_j)^2 + \sum_{i=1}^m \alpha_i,$$

implicando também que $\inf_{w,b,\xi} L(w, b, \xi, \alpha, \beta) = -\infty$. Portanto, a função dual é dada por

$$\inf_{w,b,\xi} L(w, b, \xi, \alpha, \beta) = \begin{cases} -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i, & \text{se } \begin{cases} \sum_{i=1}^m \alpha_i y_i = 0 \\ \text{e } \alpha + \beta = Ce. \end{cases} \\ -\infty, & \text{se } \begin{cases} \sum_{i=1}^m \alpha_i y_i \neq 0 \\ \text{ou } \alpha_j + \beta_j \neq C. \end{cases} \end{cases}$$

Deste modo, o problema dual (2.42) pode ser reescrito como

$$\begin{aligned}
 \max_{\alpha, \beta} \quad & -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i \\
 \text{s.a} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\
 & \alpha_i + \beta_i = C, \quad i = 1, \dots, m, \\
 & \alpha \geq 0, \beta \geq 0.
 \end{aligned} \tag{2.43}$$

Note que se (α^*, β^*) é solução do problema (2.43), então α^* é solução do problema

$$\begin{aligned}
 \max_{\alpha} \quad & -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \sum_{i=1}^m \alpha_i \\
 \text{s.a} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\
 & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m,
 \end{aligned} \tag{2.44}$$

pois a função objetivo é a mesma, dependendo apenas de α , e $0 \leq \alpha_i^* \leq \alpha_i^* + \beta_i^* = C$, já que $\beta_i^* \geq 0$. Reciprocamente, se α^* é uma solução do problema (2.44), definindo $\beta^* = Ce - \alpha^*$, concluímos que (α^*, β^*) é solução do problema (2.43).

Desta forma, a formulação dual dada em (2.44) é mais conveniente pois não envolve a variável β . Note que a importância do dual se dá também pela sua dimensão: no caso da margem rígida a dimensão do problema primal era $n + 1$ e a dimensão do dual era m ; no caso da margem flexível, a dimensão do problema primal é $n + 1 + m$ e a dimensão do dual continua sendo m .

Vejamos agora que o problema dual (2.44) de fato possui solução.

Teorema 2.18 *O problema (2.44) possui maximizador global e o gap de dualidade é zero.*

Demonstração. Pelo Teorema 2.13, o problema primal (2.17) possui um minimizador global (w^*, b^*, ξ^*) . Como as restrições deste problema são lineares, elas satisfazem uma condição de qualificação. Portanto, existem multiplicadores $\alpha^*, \beta^* \in \mathbb{R}_+^m$ tais que $(w^*, b^*, \xi^*, \alpha^*, \beta^*)$ satisfaz as condições de KKT para o problema (2.17). Pelo Teorema 1.40, concluímos que (α^*, β^*) é maximizador global do problema dual (2.42), e portanto de (2.43), e que não há *gap* de dualidade. Pela discussão acima, também temos que α^* é solução global de (2.44). \square

Podemos agora discutir como obter uma solução do primal a partir de uma solução do dual. Considere $\alpha^* \in \mathbb{R}^m$ um maximizador global arbitrário do problema dual (2.44). Definindo $\beta^* = Ce - \alpha^*$, temos que (α^*, β^*) é solução do problema dual (2.43). Assim, pelo Teorema 1.40, basta encontrar (w^*, b^*, ξ^*) de modo que $(w^*, b^*, \xi^*, \alpha^*, \beta^*)$ satisfaça

as condições de KKT para o problema primal (2.17), a saber

$$\begin{pmatrix} w \\ 0 \\ Ce \end{pmatrix} - \begin{pmatrix} \sum_{i=1}^m \alpha_i y_i x^i \\ \sum_{i=1}^m \alpha_i y_i \\ \alpha + \beta \end{pmatrix} = 0 \quad (2.45)$$

$$\alpha_i (1 - \xi_i - y_i ((w)^T x^i + b)) = 0, \quad i = 1, \dots, m, \quad (2.46)$$

$$\beta_i \xi_i = 0, \quad i = 1, \dots, m, \quad (2.47)$$

$$1 - \xi_i - y_i ((w)^T x^i + b) \leq 0, \quad i = 1, \dots, m, \quad (2.48)$$

$$\alpha_i \geq 0, \quad \beta_i \geq 0, \quad \xi_i \geq 0, \quad i = 1, \dots, m, \quad (2.49)$$

onde e representa o vetor com todas as componentes iguais a 1.

Definindo

$$w^* = \sum_{i=1}^m \alpha_i^* y_i x^i = \sum_{i \in I} \alpha_i^* y_i x^i, \quad (2.50)$$

onde $I = \{i \mid \alpha_i^* > 0\}$, obtemos (2.45). A questão agora é como obter b^* e ξ^* de modo que as condições (2.46)–(2.49) sejam satisfeitas. Isto será feito considerando dois casos, conforme os resultados a seguir.

Teorema 2.19 *Suponha que existe $j \in \{1, \dots, m\}$ tal que $0 < \alpha_j^* < C$. Então, o problema primal (2.17) possui uma única solução (w^*, b^*, ξ^*) , onde w^* é definido por (2.50),*

$$b^* = y_j - (w^*)^T x^j \quad e \quad \xi_i^* = \begin{cases} 1 - y_i ((w^*)^T x^i + b^*), & \text{se } i \in I \\ 0, & \text{se } i \notin I. \end{cases}$$

Demonstração. Pelo Teorema 1.38, se (w^*, b^*, ξ^*) é solução do problema primal (2.17), então $(w^*, b^*, \xi^*, \alpha^*, \beta^*)$ satisfaz (2.45)–(2.49). Por (2.45), concluímos que w^* cumpre (2.50). Além disso, de $\beta_j^* = C - \alpha_j^* > 0$ e (2.47) segue que $\xi_j^* = 0$. Assim, como $\alpha_j^* > 0$, obtemos de (2.46) que $1 - y_j ((w^*)^T x^j + b^*) = 0$, donde segue que

$$b^* = \frac{1}{y_j} - (w^*)^T x^j = y_j - (w^*)^T x^j.$$

Finalmente, para $i \in I$ temos $\alpha_i^* > 0$ e assim $\xi_i^* = 1 - y_i ((w^*)^T x^i + b^*)$. Para $i \notin I$ temos $\beta_i^* = C - \alpha_i^* = C > 0$, donde segue que $\xi_i^* = 0$. Pelo Teorema 2.14 a componente w é única e portanto b e ξ também são únicos. \square

Teorema 2.20 *Suponha que toda solução do problema dual (2.44) satisfaz $\alpha_i = 0$ ou $\alpha_i = C$ para todo $i = 1, \dots, m$. Então este problema dual tem uma única solução α^* . Além disso, definindo $\beta^* = Ce - \alpha^*$ e $I = \{i \mid \alpha_i^* = C\}$, a(s) solução(ões) do problema primal*

(2.17) é(são) o(s) vetor(es) (w^*, b^*, ξ^*) , onde w^* é dado por (2.50) e b^* e ξ^* satisfazem

$$\begin{cases} y_i((w^*)^T x^i + b^*) + \xi_i^* = 1, & \text{se } i \in I \\ y_i((w^*)^T x^i + b^*) \geq 1, & \text{se } i \notin I \\ \xi_i^* = 0, & i \notin I \\ \xi^* \geq 0. \end{cases} \quad (2.51)$$

Demonstração. Note primeiro que se existissem duas soluções distintas do dual, como o conjunto das soluções é convexo, poderíamos tomar uma outra solução com alguma componente no intervalo $(0, C)$, obtida como uma combinação convexa. Portanto, o dual tem uma única solução, que será denotada por α^* . Pelo Teorema 1.38, se (w^*, b^*, ξ^*) é solução do problema primal (2.17), então $(w^*, b^*, \xi^*, \alpha^*, \beta^*)$ satisfaz (2.45)–(2.49). Portanto, w^* cumpre (2.50). Além disso, para $i \in I$ temos por (2.46) que $1 - \xi_i^* - y_j((w^*)^T x^j + b^*) = 0$ e para $i \notin I$ temos por (2.45) que $\beta_i^* = C$, o que implica por (2.47) que $\xi_i^* = 0$. Desta forma, concluímos que (2.51) se verifica. Reciprocamente, podemos ver que se um vetor (w^*, b^*, ξ^*) cumpre (2.50) e (2.51), então (2.45)–(2.49) serão satisfeitas e portanto (w^*, b^*, ξ^*) é solução de (2.17). \square

Uma vez obtidos os parâmetros do classificador por meio do Teorema 2.19 ou do Teorema 2.20, podemos classificar um novo dado x usando a função de decisão dada por

$$F(x) = \text{senal} \left(\sum_{i \in I} \alpha_i^* y_i (x^i)^T x + b^* \right),$$

com $I = \{i \mid \alpha_i^* > 0\}$.

Uma consequência imediata dos Teoremas 2.19 e 2.20 é que se o problema primal (2.17) não tem solução única, então o dual (2.44) tem uma única solução α^* que satisfaz $\alpha_i^* = 0$ ou $\alpha_i^* = C$ para todo $i = 1, \dots, m$.

Outro fato que deve ser notado é que o Teorema 2.20 não representa uma recíproca para o Teorema 2.19, pois não garante que existirão soluções distintas para o problema primal, caso $\alpha_i^* = 0$ ou $\alpha_i^* = C$ para todo $i = 1, \dots, m$. Isto é confirmado com o seguinte exemplo.

Exemplo 2.21 Considere o conjunto de dados

$$X = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right\}$$

com saídas $Y = \{1, 1, 1, -1, -1\}$ e o problema primal (2.17) com $C = 1$. Note primeiro

$$\text{que } w^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, b^* = 1, \xi^* = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 2 \end{pmatrix}, \alpha^* = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \text{ e } \beta^* = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{ cumprem as}$$

condições (2.45)–(2.49). Portanto, (w^*, b^*, ξ^*) é solução do problema primal e (α^*, β^*) é solução do dual. Suponha agora que (w', b', ξ') também é solução do primal. Então $w' = w^*$ e pelo Teorema 1.40 temos que $(w^*, b', \xi', \alpha^*, \beta^*)$ satisfaz (2.45)–(2.49). Isto implica em particular que $1 - \xi'_1 - b' = 0$ e $1 - \xi'_3 - b' \leq 0$. Como $\beta^*_3 = 1$, temos $\xi'_3 = 0$ e consequentemente

$$1 - b' = \xi'_1 \geq 0 \quad \text{e} \quad 1 - b' \leq \xi'_3 = 0$$

fornecendo $b' = 1$. Pela complementaridade (2.46), $\xi'_1 = \xi'_2 = 0$ e $\xi'_4 = \xi'_5 = 2$, provando assim que o primal tem solução única. Vejamos agora que o mesmo vale para o dual. Considere então $\alpha' \in \mathbb{R}^m$ uma solução do problema dual (2.44). Definindo $\beta' = e - \alpha'$, temos que (α', β') é solução do problema dual (2.43). Assim, pelo Teorema 1.40 temos que $(w^*, b^*, \xi^*, \alpha', \beta')$ satisfaz (2.45)–(2.49). Como $\xi^*_4 = \xi^*_5 = 2$, temos $1 - \alpha'_4 = \beta'_4 = 0$ e $1 - \alpha'_5 = \beta'_5 = 0$. Portanto, de (2.45) segue que $(\alpha'_1, \alpha'_2, \alpha'_3)$ é uma solução do sistema

$$\begin{cases} \alpha_1 + 2\alpha_2 + 3\alpha_3 = 3 \\ \alpha_1 - \alpha_2 + \alpha_3 = 0 \\ \alpha_1 + \alpha_2 + \alpha_3 = 2, \end{cases}$$

donde segue que $\alpha' = \alpha^*$ e $\beta' = \beta^*$.

A definição de vetor suporte no caso de margem flexível é dada em termos de uma solução do problema dual.

Definição 2.22 Considere α^* uma solução do problema dual (2.44). Os vetores suporte são aqueles tais que $\alpha^*_i > 0$.

Note que pelo Teorema 2.19 temos $\{x^j \in X \mid 0 < \alpha^*_j < C\} \subset H_1 \cup H_2$. Tais vetores são conhecidos como vetores suporte *bound*. Os outros vetores suporte são os que tem $\alpha^*_i = C$, implicando em $\xi^* = 1 - y_i((w^*)^T x^i + b^*)$. Veja que este é o único caso que as variáveis de folga podem ser diferentes de zero. Assim, x^i está classificado incorretamente se $\xi^*_i \geq 1$, quando $\xi^*_i = 1$, temos x^i sobre o hiperplano ótimo $(w^*)^T x + b^* = 0$. Se $0 < \xi^*_i < 1$, x^i é um ponto corretamente classificado, conhecido como vetor suporte *non-bound*. Por fim, quando $\xi^*_i = 0$, temos x^i sobre um dos hiperplanos H_1 ou H_2 .

Outro fato sobre os vetores suporte é dado no seguinte resultado e diz que se o problema primal tem mais do que uma solução, então pelo menos um dos dois hiperplanos não contem vetor suporte.

Teorema 2.23 *Seja (w^*, b^*, ξ^*) uma solução do problema primal (2.17) e α^* uma solução do problema dual (2.44). Considere $I = \{i \mid \alpha_i^* > 0\}$ e os conjuntos $N_i, i = 1, \dots, 6$ definidos no Teorema 2.17. Se $|N_2 \cup N_4| = |N_1|$, então*

$$I = N_1 \cup N_2 \cup N_4.$$

Isto é, nenhum vetor suporte está sobre o hiperplano H_1 e todos os vetores x^i que estão em H_2 são vetores suporte. Se $|N_1 \cup N_3| = |N_2|$, então

$$I = N_1 \cup N_2 \cup N_3,$$

o que significa que nenhum vetor suporte está sobre o hiperplano H_2 e todos os vetores x^i que estão em H_1 são vetores suporte.

Demonstração. Suponha que $|N_2 \cup N_4| = |N_1|$. Como $\alpha_i = 0$ para $i \in N_5 \cup N_6$, obtemos de (2.45) que

$$\sum_{i \in N_1} \alpha_i - \sum_{i \in N_2} \alpha_i + \sum_{i \in N_3} \alpha_i - \sum_{i \in N_4} \alpha_i = \sum_{i=1}^m \alpha_i y_i = 0 \quad (2.52)$$

Para os índices $i \in N_1 \cup N_2$ temos $\xi_i^* > 0$ e assim $\alpha_i = C$, pois $(C - \alpha_i^*)\xi_i^* = \beta_i^*\xi_i^* = 0$. Portanto podemos reescrever (2.52) como

$$C|N_1| + \sum_{i \in N_3} \alpha_i = C|N_2| + \sum_{i \in N_4} \alpha_i = C|N_2 \cup N_4| + \sum_{i \in N_4} (\alpha_i - C),$$

donde segue que

$$0 \leq \sum_{i \in N_3} \alpha_i = \sum_{i \in N_4} (\alpha_i - C) \leq 0,$$

implicando por sua vez em $\alpha_i = 0$ para todo $i \in N_3$ e $\alpha_i = C$ para todo $i \in N_4$. A outra afirmação se prova de modo análogo. \square

Note que os pontos x^i que não são vetores suporte, isto é, os que cumprem $\alpha_i^* = 0$, satisfazem $\xi_i^* = 0$. Portanto, por (2.48) temos que $y_i ((w^*)^T x^i + b^*) \geq 1$, implicando que x^i está no semiespaço correto definido pela margem. Novamente, podemos ter pontos x^i em H_1 ou H_2 associados a $\alpha_i^* = 0$.

Uma consequência do Teorema 2.23 é que se o problema primal tem mais do que uma solução, então a quantidade de vetores suporte da classe positiva é a mesma que da classe negativa. Além disso, existe uma solução para a qual a folga associada a cada vetor suporte é estritamente positiva.

Corolário 2.24 *Suponha que o problema primal (2.17) tem mais do que uma solução. Então o dual possui uma única solução α^* e $\sum_{i \in I} y_i = 0$, onde $I = \{i \mid \alpha_i^* > 0\}$. Além disso, existe uma solução $(w^*, \tilde{b}, \tilde{\xi})$ tal que $\tilde{\xi}_i > 0$ para todo $i \in I$.*

Demonstração. Como a hipótese do Teorema 2.19 não se cumpre, podemos aplicar o

Teorema 2.20 para concluir que o dual possui solução única. Considere agora duas soluções distintas do primal, (w^*, b^*, ξ^*) e (w^*, b', ξ') , e os conjuntos $N_i, i = 1, \dots, 6$, definidos no Teorema 2.17. Por este teorema temos que $|N_2 \cup N_4| = |N_1|$ ou $|N_1 \cup N_3| = |N_2|$. Pelo Teorema 2.23, temos

$$I = N_1 \cup N_2 \cup N_4 \quad \text{ou} \quad I = N_1 \cup N_2 \cup N_3,$$

respectivamente. Portanto,

$$\sum_{i \in I} y_i = \sum_{i \in N_1} y_i + \sum_{i \in N_2 \cup N_4} y_i = 0 \quad \text{ou} \quad \sum_{i \in I} y_i = \sum_{i \in N_2} y_i + \sum_{i \in N_1 \cup N_3} y_i = 0,$$

respectivamente, provando assim a primeira afirmação. Para verificar a outra, suponha sem perda de generalidade que $b' > b^*$ e considere a solução

$$(w^*, \tilde{b}, \tilde{\xi}) = (1-t)(w^*, b^*, \xi^*) + t(w^*, b', \xi') = (w^*, b^* + \delta, \tilde{\xi})$$

com $t \in (0, 1]$ e $\delta = t(b' - b^*)$. Escolha t suficientemente pequeno de tal maneira que $(w^*)^T x^i + b^* + \delta < 1$ para todo $i \in N_1$ e $(w^*)^T x^i + b^* + \delta < -1$ para todo $i \in N_6$. Assim, $\tilde{\xi}_i = 1 - (w^*)^T x^i - b^* - \delta > 0$ para $i \in N_1$ e $\tilde{\xi}_i = 0$ para $i \in N_6$. Para $i \in N_2 \cup N_4$, o Teorema 2.15 nos permite concluir que

$$y_i((w^*)^T x^i + b^* + \delta) = y_i((w^*)^T x^i + b^*) - \delta = 1 - (\xi_i^* + \delta) < 1.$$

Portanto, este mesmo teorema implica

$$\tilde{\xi}_i = 1 - y_i((w^*)^T x^i + b^* + \delta) > 0.$$

Finalmente, se $i \in N_3 \cup N_5$, temos

$$y_i((w^*)^T x^i + b^* + \delta) = (w^*)^T x^i + b^* + \delta \geq 1 + \delta > 1,$$

donde segue que $\tilde{\xi}_i = 0$. Além disso, como $(w^*, b^* + \delta, \tilde{\xi})$ e α^* cumprem (2.46), temos $\alpha_i^* = 0$ para $i \in N_3$ e conseqüentemente $i \notin I$. \square

O corolário acima nos diz que quando o problema primal tem infinitas soluções, o conjunto de vetores suporte é unicamente determinado e podemos obter uma solução do primal com todas as folgas associadas aos vetores suporte positivas. Ou seja, nenhum vetor suporte está sobre a margem correta, diferentemente do que ocorre no caso linearmente separável, onde todos os vetores suporte estavam sobre as correspondentes margens.

Veremos na próxima seção como lidar com casos nos quais mesmo introduzindo variáveis de folga não conseguimos obter um bom classificador. Para formular tais problemas, usaremos uma ferramenta conhecida como truque do *Kernel*.

2.4 Máquinas de Vetores Suporte - Não linear

Para problemas em que o conjunto de dados não é linearmente separável, isto é, quando não é possível obter um hiperplano separador, nem mesmo usando regularização, Boser, Guyon e Vapnik [22], em 1992, sugeriram uma maneira de criar classificadores não lineares, aplicando o Truque do *Kernel*. Esta ferramenta permite mapear os dados do espaço original para um espaço de dimensão mais elevada (chamado *espaço de características*), na esperança de que neste novo espaço seja possível realizar uma classificação linear da mesma maneira que tratamos nas seções anteriores. Este procedimento de alterar a representação dos dados vem sendo muito utilizado em Aprendizagem de Máquina.

Considere então uma função $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^N$, com $N > n$ que mapeia os atributos $x^i \in \mathbb{R}^n$ do espaço de entrada para o espaço de características \mathbb{R}^N .

Nos Exemplos 2.25 e 2.26 apresentamos conjuntos de treinamento não linearmente separáveis, mas que ao realizar o mapeamento para um espaço de dimensão maior obtemos uma classificação linear em tal espaço.

Exemplo 2.25 Considere o conjunto de treinamento $X = \{0, 1, 2, 3\}$, com saídas $\{1, -1, -1, 1\}$ respectivamente. Note pela Figura 2.10, que não é possível obter um ponto no espaço de entrada \mathbb{R} que separe tais dados.

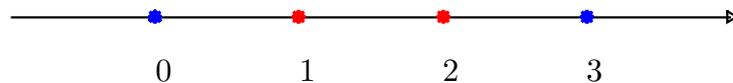


Figura 2.10: Conjunto não linearmente separável.

Tomando a função polinomial $\phi(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix}$, os atributos tornam-se linearmente separáveis no espaço de características \mathbb{R}^2 . Assim, mapeamos os dados do conjunto $X \subset \mathbb{R}$ em $\{(0, 0), (1, 1), (2, 4), (3, 9)\} \subset \mathbb{R}^2$. Agora, podemos encontrar um hiperplano separador ótimo pelo procedimento descrito na Seção 2.2, veja Figura 2.11.

Exemplo 2.26 Na Figura 2.12(a), temos um conjunto de treinamento

$$X = \{(-2, -2), (-2, 2), (-1, -1), (-1, 1), (0, 0), (1, -1), (1, 1), (2, -2), (2, 2)\}$$

com saídas $\{-1, -1, 1, 1, 1, 1, 1, -1, -1\}$ respectivamente, que não é linearmente separável.

Usando o mapeamento $\phi(x) = (x_1, x_2, x_1^2 + x_2^2)^T$ obtemos um conjunto linearmente separável em \mathbb{R}^3 , conforme vemos na Figura 2.12(b). Assim, podemos resolver o problema (2.3) trocando x^i por $\phi(x^i)$, obtendo um hiperplano ótimo em \mathbb{R}^3 , ilustrado na Figura 2.12(c). Realizando a intersecção do hiperplano ótimo em \mathbb{R}^3 com a superfície descrita por ϕ , como ilustrado na Figura 2.12(d), obtemos uma circunferência separando os dados no espaço de entrada, veja Figura 2.13(a).

Na Figura 2.13(b), a região em vermelho representa os pontos da classe negativa e a região em azul, os pontos da classe positiva. Note que a medida que os pontos ficam

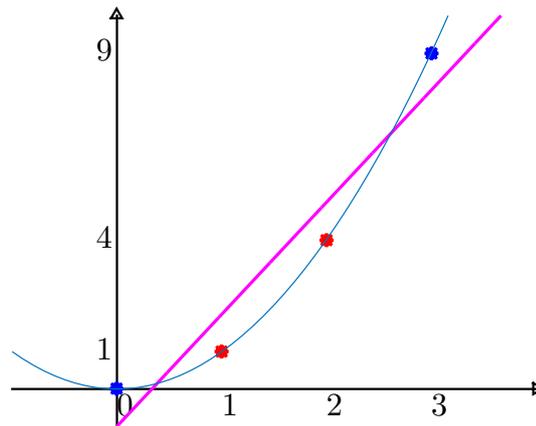
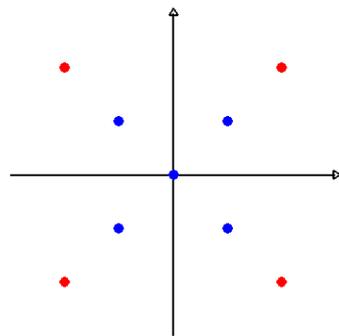
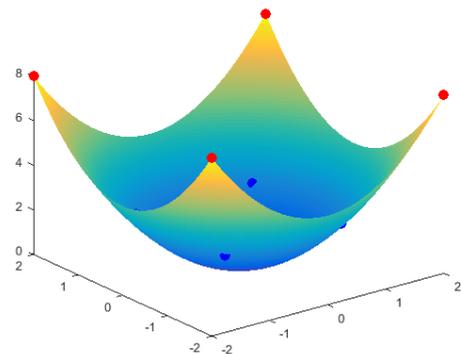


Figura 2.11: Hiperplano e curva descrita pela função ϕ , no espaço de características.

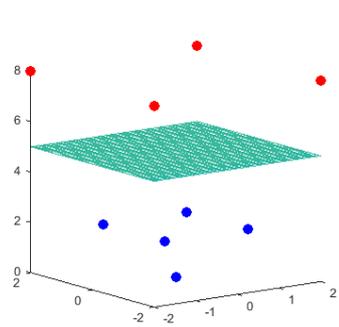
próximos da superfície separadora no espaço de entrada, as cores azul e vermelho tornam-se mais claras. Ilustramos isto para tratar da confiança na classificação dos dados, pois nas regiões mais escuras o valor da função de decisão é maior.



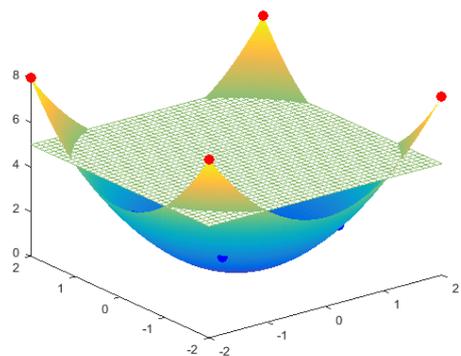
(a) Dados não linearmente separáveis.



(b) Dados mapeados em \mathbb{R}^3 .



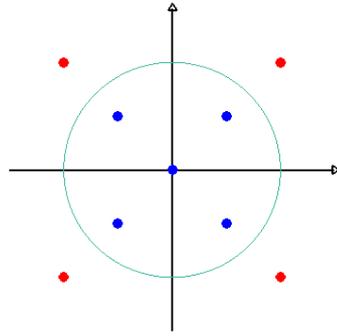
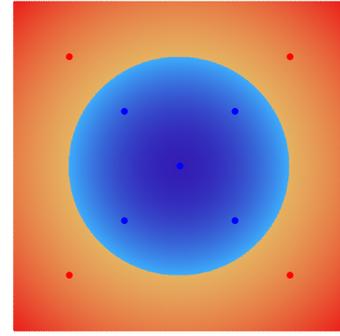
(c) Hiperplano no espaço de características.



(d) Intersecção Hiperplano e Superfície.

Figura 2.12: Obtenção do hiperplano.

Deste modo, para classificar um novo dado, podemos mapeá-lo para o espaço

(a) Superfície separadora em \mathbb{R}^2 .

(b) Regiões de classificação.

Figura 2.13: Classificação no espaço de entrada.

de características e analisar sua posição em relação ao hiperplano ótimo. Entretanto, usaremos o que já tratamos nas seções anteriores, construindo uma função de decisão que dependerá apenas das informações do espaço de entrada. A forma da superfície de separação dos dados no espaço de entrada não será um hiperplano, podendo ser uma superfície de difícil representação.

O problema primal associado aos dados mapeados no espaço de características pode ser escrito como

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 & w \in \mathbb{R}^N, b \in \mathbb{R} \\ \text{s.a} \quad & y_i(w^T \phi(x^i) + b) \geq 1, \quad i = 1, \dots, m. \end{aligned} \quad (2.53)$$

No entanto, não temos a garantia que o mapeamento escolhido realmente torna o conjunto de treinamento linearmente separável. Para lidar com esta possibilidade usaremos regularização, considerando o seguinte problema

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i & w \in \mathbb{R}^N, b \in \mathbb{R}, \xi \in \mathbb{R}^m \\ \text{s.a} \quad & y_i(w^T \phi(x^i) + b) \geq 1 - \xi_i, \quad i = 1, \dots, m, \\ & \xi_i \geq 0, & i = 1, \dots, m. \end{aligned} \quad (2.54)$$

Note que se N for muito grande, o problema acima pode ficar difícil de trabalhar computacionalmente. Assim, usaremos a formulação dual do problema (2.54), dada por

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2} \sum_{i=1}^m \alpha_i \alpha_j y_i y_j \phi(x^i)^T \phi(x^j) + \sum_{i=1}^m \alpha_i \\ \text{s.a} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m. \end{aligned} \quad (2.55)$$

Notamos que todos os resultados estabelecidos nas Seções 2.2 e 2.3, em particular os que tratam da existência e unicidade, se aplicam aqui também, bastando trocar x^i por $\phi(x^i)$. No caso do problema dual (2.55), a única diferença em relação ao problema dual (2.44) está na troca do produto interno $(x^i)^T x^j$, no espaço original, por $\phi(x^i)^T \phi(x^j)$. Assim, precisamos saber como calcular produtos internos no espaço de características e, para isso, usamos as chamadas funções *Kernel*.

Definição 2.27 *Considere um conjunto $\mathcal{X} \subset \mathbb{R}^n$. Uma função $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ é chamada *Kernel* se existe $\phi : \mathcal{X} \rightarrow \mathbb{R}^N$ tal que*

$$K(x, z) = \phi(x)^T \phi(z)$$

para todos $x, z \in \mathcal{X}$. Neste contexto, \mathbb{R}^N é chamado espaço de características e ϕ um mapeamento de \mathcal{X} em \mathbb{R}^N .

A definição de *Kernel* apresentada acima será generalizada mais adiante considerando espaços de Hilbert, mais gerais que \mathbb{R}^N .

Exemplo 2.28 *As funções $K_1, K_2 : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ dadas por*

$$K_1(x, z) = (x^T z)^2 \tag{2.56}$$

e

$$K_2(x, z) = x_1 z_1 + x_2 z_2 + x_1^2 z_1^2 + x_1^2 z_2^2 + x_2^2 z_1^2 + x_2^2 z_2^2$$

são *Kernels* associados às funções $\phi, \psi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ definidas por $\phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1 x_2 \\ x_2^2 \end{pmatrix}$

e $\psi(x) = \begin{pmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{pmatrix}$, respectivamente. O *Kernel* dado em (2.56) é conhecido como *Kernel polinomial*.

O Truque do *Kernel* consiste em substituir o produto interno $\phi(x^i)^T \phi(x^j)$ que aparece no problema (2.55) pelo número real $K(x^i, x^j)$. A vantagem é que o espaço de características não precisa ser construído explicitamente, ou seja, não precisamos conhecer sua dimensão N e nem seus elementos $\phi(x^i)$.

Em virtude da similaridade entre os problemas (2.44) e (2.55), bem como dos resultados estabelecidos nos Teoremas 2.19 e 2.20, podemos obter uma solução do problema primal (2.54) a partir de uma solução α^* do dual (2.55). Mais precisamente, temos

$$w^* = \sum_{i \in I} \alpha_i^* y_i \phi(x^i), \tag{2.57}$$

onde $I = \{i \mid \alpha_i^* > 0\}$. Além disso, caso exista $j \in \{1, \dots, m\}$ tal que $0 < \alpha_j^* < C$, temos

$$b^* = y_j - (w^*)^T \phi(x^j) \quad \text{e} \quad \xi_i^* = \begin{cases} 1 - y_i((w^*)^T \phi(x^i) + b^*), & i \in I \\ 0, & i \notin I, \end{cases} \quad (2.58)$$

resultando assim que o primal (2.54) tem uma única solução. Por outro lado, caso $\alpha_i^* = 0$ ou $\alpha_i^* = C$ para todo $i = 1, \dots, m$, podemos obter b^* e ξ^* (não necessariamente únicos) satisfazendo

$$\begin{cases} y_i((w^*)^T \phi(x^i) + b^*) + \xi_i^* = 1, & \text{se } i \in I \\ y_i((w^*)^T \phi(x^i) + b^*) \geq 1, & \text{se } i \notin I \\ \xi_i^* = 0, & i \notin I \\ \xi^* \geq 0. \end{cases} \quad (2.59)$$

Veja que o vetor w^* dado em (2.57) pode não ser obtido de forma explícita, pois seu cálculo envolve a função ϕ que nem sempre é conhecida. Do mesmo modo, o cálculo de b^* e ξ^* por meio de (2.58) ou (2.59) também depende (a princípio) de ϕ e de produtos internos em \mathbb{R}^N . Entretanto, veremos agora que a função de decisão pode ser obtida sem o conhecimento de ϕ e de N . De fato, note primeiro que podemos reescrever (2.58) e (2.59) como

$$b^* = y_j - \sum_{\ell \in I} \alpha_\ell^* y_\ell K(x^\ell, x^j) \quad \text{e} \quad \xi_i^* = \begin{cases} 1 - y_i \left(\sum_{\ell \in I} \alpha_\ell^* y_\ell K(x^\ell, x^i) + b^* \right), & i \in I \\ 0, & i \notin I, \end{cases}$$

e

$$\begin{cases} y_i \left(\sum_{\ell \in I} \alpha_\ell^* y_\ell K(x^\ell, x^i) + b^* \right) + \xi_i^* = 1, & \text{se } i \in I \\ y_i \left(\sum_{\ell \in I} \alpha_\ell^* y_\ell K(x^\ell, x^i) + b^* \right) \geq 1, & \text{se } i \notin I \\ \xi_i^* = 0, & i \notin I \\ \xi^* \geq 0. \end{cases} \quad (2.60)$$

respectivamente. Em particular, b^* pode ser obtido apenas com o conhecimento da função *Kernel* e dos dados no espaço de entrada. Desta forma, a função de decisão pode ser escrita como

$$F(x) = \text{sinal}((w^*)^T \phi(x) + b^*) = \text{sinal} \left(\sum_{i \in I} \alpha_i^* y_i K(x^i, x) + b^* \right), \quad (2.61)$$

que também depende apenas dos dados no espaço de entrada e da função *Kernel*, sem precisar portanto conhecer o mapeamento para o espaço de características.

Observação 2.29 *Sentimos falta na literatura, por exemplo, em [3, 15, 20], da discussão sobre o cálculo de b^* quando não temos unicidade. Desta forma, uma contribuição nossa neste contexto consiste em analisar tal situação. Ou seja, quando b^* é calculado por meio*

de (2.60). Isto significa que podemos ter infinitas funções de decisão. Neste caso, pontos podem ser considerados da classe positiva de acordo com uma delas e classificados como da classe negativa segundo outra função de decisão.

Vamos agora discutir com mais detalhes as propriedades e a caracterização das funções *Kernel*.

Para tanto, vamos considerar espaços de características mais gerais que \mathbb{R}^N , a saber os espaços de Hilbert, nos quais temos um produto interno $\langle \cdot, \cdot \rangle$. Desta forma, a Definição 2.27 pode ser reformulada como segue.

Definição 2.30 Considere um conjunto $\mathcal{X} \subset \mathbb{R}^n$. Uma função $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ é chamada *Kernel* se existe um mapeamento $\phi : \mathcal{X} \rightarrow \mathcal{H}$, do conjunto \mathcal{X} em um espaço de Hilbert \mathcal{H} , tal que

$$K(x, z) = \langle \phi(x), \phi(z) \rangle$$

para todos $x, z \in \mathcal{X}$.

Note que qualquer função que cumpre a Definição 2.27 também será um *Kernel* de acordo com a Definição 2.30 pois \mathbb{R}^N é um espaço de Hilbert. Em particular as funções do Exemplo 2.28 continuam sendo *Kernels* aqui. A recíproca, no entanto, não é verdadeira, conforme podemos ver no exemplo seguinte.

Exemplo 2.31 Considere o espaço ℓ_2 , das seqüências de quadrado somável, dado no Exemplo 1.12. A função $K : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ dada por $K(x, z) = e^{xz}$ é *Kernel* associado à função $\phi : \mathbb{R} \rightarrow \ell_2$ dada por $\phi(x) = \left(\frac{x^k}{\sqrt{k!}} \right)_{k \in \mathbb{N}}$. Note que ϕ está bem definida pois

$$\sum_{k=0}^{\infty} (\phi_k(x))^2 = \sum_{k=0}^{\infty} \frac{(x^2)^k}{k!} = e^{x^2} < \infty.$$

Por conveniência, consideramos aqui $\mathbb{N} = \{0, 1, 2, \dots\}$.

Proposição 2.32 Não existe $\phi : \mathbb{R} \rightarrow \mathbb{R}^N$, com $N \in \mathbb{N}$, tal que $e^{xz} = \phi(x)^T \phi(z)$ para todos $x, z \in \mathbb{R}$.

Demonstração. Suponha por absurdo que exista $\phi : \mathbb{R} \rightarrow \mathbb{R}^N$, com $N \in \mathbb{N}$, tal que $e^{xz} = \phi(x)^T \phi(z)$ para todos $x, z \in \mathbb{R}$. Considere $m > N$ e defina $A = (\phi(1) \cdots \phi(m)) \in \mathbb{R}^{N \times m}$. Assim,

$$A^T A = \begin{pmatrix} \phi(1)^T \\ \vdots \\ \phi(m)^T \end{pmatrix} (\phi(1) \cdots \phi(m)) = \begin{pmatrix} e & e^2 & \cdots & e^m \\ e^2 & e^4 & \cdots & e^{2m} \\ \vdots & \vdots & \ddots & \vdots \\ e^m & e^{2m} & \cdots & e^{m^2} \end{pmatrix}.$$

Como $A^T A$ é uma matriz de Vandermonde, temos que

$$\det(A^T A) = e^s \prod_{k>j} (e^k - e^j) \neq 0,$$

onde $s = \frac{m(m+1)}{2}$. Portanto, $\text{posto}(A^T A) = m$, contradizendo o fato de

$$\text{posto}(A^T A) = \text{posto}(A) \leq N < m.$$

□

Uma observação importante é que o mapeamento ϕ que define uma função *Kernel* não é único. Em [8] temos um exemplo que por comodidade reproduzimos a seguir.

Exemplo 2.33 Considere a função $K : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ dada por $K(x, z) = (x^T z)^2$. Conforme vimos no Exemplo 2.28 esta função é *Kernel* pois $K(x, z) = \phi(x)^T \phi(z)$, com $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ dada por

$$\phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}.$$

No entanto, usando $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ ou $\zeta : \mathbb{R}^2 \rightarrow \mathbb{R}^4$ dadas por

$$\psi(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} x_1^2 - x_2^2 \\ 2x_1x_2 \\ x_1^2 + x_2^2 \end{pmatrix} \quad e \quad \zeta(x) = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{pmatrix},$$

temos também $K(x, z) = \psi(x)^T \psi(z) = \zeta(x)^T \zeta(z)$.

Uma questão natural que surge agora é saber quais são as condições para que uma dada função $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ seja *Kernel*, sendo assim admissível para uso em SVM. Os dois resultados a seguir mostram algumas condições necessárias, clássicas na literatura [15].

Teorema 2.34 Considere um conjunto $\mathcal{X} \subset \mathbb{R}^n$ e suponha que $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ seja *Kernel*. Então,

- (a) K é simétrica, no sentido de que $K(x, z) = K(z, x)$;
- (b) Vale a desigualdade de Cauchy-Schwarz generalizada

$$K(x, z)^2 \leq K(x, x)K(z, z).$$

Demonstração. De fato, temos

$$K(x, z) = \langle \phi(x), \phi(z) \rangle = \langle \phi(z), \phi(x) \rangle = K(z, x)$$

e

$$K(x, z)^2 = \langle \phi(x), \phi(z) \rangle^2 \leq \|\phi(x)\|^2 \|\phi(z)\|^2 = K(x, x)K(z, z).$$

□

É fácil ver que as condições do Teorema 2.34 não são suficientes. De fato, a função $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ dada por $K(x, z) = -x^T z$ cumpre as duas condições, mas não podemos escrever $K(x, z) = \langle \phi(x), \phi(z) \rangle$, pois isto implicaria

$$0 \leq \|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle = K(x, x) = -\|x\|^2 \leq 0.$$

Teorema 2.35 *Se $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ é Kernel, então a matriz de Gram associada a qualquer conjunto finito $X = \{x^1, \dots, x^m\} \subset \mathcal{X}$, definida por $G \in \mathbb{R}^{m \times m}$, $G(i, j) = K(x^i, x^j)$, é simétrica semidefinida positiva.*

Demonstração. Denotando $a_{ij} = G(i, j) = \langle \phi(x^i), \phi(x^j) \rangle$ e $v = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} \in \mathbb{R}^m$, temos

$$v^T G v = \sum_{i,j=1}^m a_{ij} c_i c_j = \sum_{i,j=1}^m \langle c_i \phi(x^i), c_j \phi(x^j) \rangle = \left\langle \sum_{i=1}^m c_i \phi(x^i), \sum_{j=1}^m c_j \phi(x^j) \right\rangle \geq 0.$$

□

O interessante agora é que se o conjunto \mathcal{X} for compacto e K for contínua, então vale a recíproca do Teorema 2.35.

Teorema 2.36 *Sejam $\mathcal{X} \subset \mathbb{R}^n$ um conjunto compacto e $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ uma função contínua. Suponha que para todo conjunto finito $X = \{x^1, \dots, x^m\} \subset \mathcal{X}$ a matriz de Gram associada seja simétrica semidefinida positiva. Então K é Kernel, mais precisamente, existe uma função $\phi : \mathcal{X} \rightarrow \ell_2$ tal que*

$$K(x, z) = \langle \phi(x), \phi(z) \rangle = \sum_{k=1}^{\infty} \phi_k(x) \phi_k(z).$$

A prova deste resultado não será apresentada aqui pois envolve conceitos de Análise Funcional, que estão fora do escopo deste trabalho.

Outra forma de estabelecer condições suficientes é por meio do clássico Teorema de Mercer, apresentado a seguir. Novamente, não discutiremos a demonstração por estar fora dos objetivos desta dissertação. Mais detalhes podem ser encontrados em [10, 15].

Dado um conjunto compacto $\mathcal{X} \subset \mathbb{R}^n$, considere o espaço

$$L_2(\mathcal{X}) = \left\{ f : \mathcal{X} \rightarrow \mathbb{R} \mid \int_{\mathcal{X}} (f(x))^2 dx < \infty \right\}.$$

Teorema 2.37 (Mercer) *Sejam $\mathcal{X} \subset \mathbb{R}^n$ um conjunto compacto e $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ uma função contínua e simétrica tal que o operador integral $T_K : L_2(\mathcal{X}) \rightarrow L_2(\mathcal{X})$ dado por*

$$T_K(f)(\cdot) = \int_{\mathcal{X}} K(\cdot, z)f(z)dz$$

é semidefinido positivo, ou seja,

$$\int_{\mathcal{X} \times \mathcal{X}} K(x, z)f(x)f(z)dx dz \geq 0, \quad (2.62)$$

para todo $f \in L_2(\mathcal{X})$. Então, existe uma função $\phi : \mathcal{X} \rightarrow \ell_2$ tal que

$$K(x, z) = \langle \phi(x), \phi(z) \rangle = \sum_{k=1}^{\infty} \phi_k(x)\phi_k(z).$$

É interessante notar a relação entre a positividade do operador definido pela função simétrica K e a positividade de matrizes. Isto pode ser visto considerando o caso no qual o conjunto \mathcal{X} é finito, digamos $\mathcal{X} = \{1, \dots, m\}$. Desta forma, a função K se identifica

com a matriz $(K(i, j))_{i, j \in \{1, \dots, m\}} \in \mathbb{R}^{m \times m}$ e a função f com o vetor $f = \begin{pmatrix} f(1) \\ \vdots \\ f(m) \end{pmatrix} \in \mathbb{R}^m$.

Neste caso, o operador T_K fica definido por

$$T_K(f) = Kf = \sum_{j=1}^m f(j)K(:, j).$$

Dizer então que T_K é semidefinido positivo significa

$$\sum_{i, j=1}^m f(i)f(j)K(i, j) = f^T Kf \geq 0.$$

Ainda convém observar que neste caso a prova do teorema de Mercer segue da decomposição espectral de matrizes simétricas. De fato, como K é simétrica e semidefinida positiva, existem matrizes $P, D \in \mathbb{R}^{m \times m}$ com $D = \text{diag}(\lambda_1, \dots, \lambda_m)$, $\lambda_j \geq 0$, satisfazendo $K = PDP^T$. Fazendo $U = \sqrt{D}P^T$, temos $K = U^T U$. Defina então $\phi : \mathcal{X} \rightarrow \ell_2$ e $\psi : \mathcal{X} \rightarrow \mathbb{R}^m$ por $\phi(j) = \begin{pmatrix} u^j \\ 0 \end{pmatrix}$ e $\psi(j) = u^j$, onde u^j é a j -ésima coluna de U . Assim,

$$K(i, j) = \langle \phi(i), \phi(j) \rangle = \psi(i)^T \psi(j),$$

provando não apenas o Teorema de Mercer como também que K é *Kernel* de acordo com a Definição 2.27.

Também cabe ressaltar que mesmo quando o conjunto \mathcal{X} não é finito ainda pode-

mos estabelecer relações entre a positividade dada em (2.62) e a positividade das matrizes de Gram.

Teorema 2.38 *Sejam $\mathcal{X} \subset \mathbb{R}^n$ um conjunto compacto e $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ uma função contínua e simétrica. Então,*

$$\int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz \geq 0,$$

para todo $f \in L_2(\mathcal{X})$ se, e somente se, a matriz de Gram associada a qualquer conjunto finito $X = \{x^1, \dots, x^m\} \subset \mathcal{X}$ é simétrica semidefinida positiva.

Demonstração. Segue direto do Corolário 1.14 e do Teorema 1.15. □

A hipótese de compacidade no Teorema 2.37 pode ser muito forte do ponto de vista teórico. Entretanto, para fins práticos esta restrição não é um obstáculo, pois os dados para problemas reais normalmente são limitados. Por exemplo, em um problema para prever se uma pessoa possui determinada doença, os dados são vetores com suas componentes representando glicemia, pressão arterial, colesterol, etc. Outro exemplo pode ser acerca da previsão se um cliente será adimplente ou inadimplente, em que os dados são vetores com entradas: idade, renda, etc. Nestes exemplos, as componentes dos vetores são limitadas.

Veremos agora outras propriedades e algumas operações envolvendo *Kernels*. Essas operações são muito úteis para verificar diretamente se uma função é, ou não, um *Kernel* e, adicionalmente, para construir novos *Kernels* a partir de outros [16].

Proposição 2.39 *Sejam $\mathcal{X} \subset \mathbb{R}^n$, $K, K_1, K_2 : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ funções Kernel com mapeamentos $\phi, \phi_1, \phi_2 : \mathcal{X} \rightarrow \mathcal{H}$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $a \in \mathbb{R}^+$ e $k \in \mathbb{N}$. Então as funções seguintes são Kernels.*

- (a) aK .
- (b) $K_1 + K_2$.
- (c) $K_1 K_2$, assumindo que \mathcal{X} é compacto e K_1 e K_2 são contínuas.
- (d) K^k , assumindo que \mathcal{X} é compacto e K é contínua.
- (e) $(x, z) \mapsto f(x)K(x, z)f(z)$.

Demonstração. (a) Basta tomar o mapeamento $\sqrt{a}\phi$. De fato, $aK(x, z) = \langle \sqrt{a}\phi(x), \sqrt{a}\phi(z) \rangle$.
 (b) Considere o mapeamento $\psi = (\phi_1, \phi_2) : \mathcal{X} \rightarrow \mathcal{H} \times \mathcal{H}$. Assim,

$$\begin{aligned} K_1(x, z) + K_2(x, z) &= \langle \phi_1(x), \phi_1(z) \rangle + \langle \phi_2(x), \phi_2(z) \rangle \\ &= \langle (\phi_1(x), \phi_2(x)), (\phi_1(z), \phi_2(z)) \rangle \\ &= \langle \psi(x), \psi(z) \rangle. \end{aligned}$$

(c) Dado um conjunto finito $X = \{x^1, \dots, x^m\} \subset \mathcal{X}$, considere G_1 e G_2 as matrizes de Gram associadas a K_1 e K_2 , respectivamente. Desta forma, a matriz de Gram associada a $K_1 K_2$ é o produto de Hadamard $G_1 \circ G_2$. O resultado segue então dos Teoremas 2.35, 1.7 e 2.36.

(d) Segue imediatamente de (c).

(e) Tome o mapeamento $f\phi$. Assim, $f(x)K(x, z)f(z) = \langle f(x)\phi(x), f(z)\phi(z) \rangle$. \square

Proposição 2.40 *Sejam $\mathcal{X} \subset \mathbb{R}^n$ compacto e $K_p : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, $p = 1, 2, 3, \dots$, funções Kernel. Suponha que para todos $x, z \in \mathcal{X}$ exista o limite*

$$\lim_{p \rightarrow \infty} K_p(x, z) \stackrel{\text{def}}{=} K(x, z)$$

e que a função $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ assim definida seja contínua. Então K é Kernel.

Demonstração. Dados um conjunto finito $X = \{x^1, \dots, x^m\} \subset \mathcal{X}$ e $p \in \mathbb{N}$, considere G_p a matriz de Gram associada a K_p e G a matriz de Gram associada a K . Note primeiro que G é simétrica pois

$$G(i, j) = K(x^i, x^j) = \lim_{p \rightarrow \infty} K_p(x^i, x^j) = \lim_{p \rightarrow \infty} K_p(x^j, x^i) = K(x^j, x^i) = G(j, i).$$

Além disso, considerando $v = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} \in \mathbb{R}^m$ arbitrário, temos pelo Teorema 2.35,

$$\sum_{i, j=1}^m K_p(x^i, x^j) c_i c_j = v^T G_p v \geq 0.$$

Tomando o limite quando $p \rightarrow \infty$, obtemos

$$v^T G v = \sum_{i, j=1}^m K(x^i, x^j) c_i c_j \geq 0,$$

mostrando que G é semidefinida positiva. O resultado segue então do Teorema 2.36. \square

Proposição 2.41 *Sejam $\mathcal{X} \subset \mathbb{R}^n$ compacto e $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ Kernel. Suponha também que K é contínua. Então $\exp(K) = e^K$ é Kernel.*

Demonstração. Para cada $p = 0, 1, 2, \dots$, defina $K_p(x, z) = \sum_{k=0}^p \frac{(K(x, z))^k}{k!}$. Assim, $\exp(K(x, z)) = \lim_{p \rightarrow \infty} K_p(x, z)$ e a conclusão segue das Proposições 2.39 e 2.40. \square

Para decidir se uma dada função é Kernel pode ser muito difícil verificar a validade do Teorema 2.37, pois precisamos mostrar que toda função $f \in L_2(\mathcal{X})$ deve satisfazer

(2.62). Assim, o Teorema 2.36 e as propriedades apresentadas nas Proposições 2.39, 2.40 e 2.41 podem facilitar muito esta tarefa. Por exemplo, considere a função dada por

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right),$$

onde $\sigma > 0$. Notando que

$$K(x, z) = \exp\left(-\frac{1}{2\sigma^2}\|x\|^2\right) \exp\left(\frac{1}{\sigma^2}x^T z\right) \exp\left(-\frac{1}{2\sigma^2}\|z\|^2\right),$$

concluimos que K é *Kernel*. Este *Kernel* é muito utilizado e conhecido na literatura como *Kernel Gaussiano* ou *Radial Basis Function* (RBF).

Outro exemplo consiste no chamado *Kernel* polinomial $(x^T z + p)^d$, onde $p \geq 0$ e $d \in \mathbb{N}$. Note que $x^T z + p = \begin{pmatrix} x \\ \sqrt{p} \end{pmatrix}^T \begin{pmatrix} z \\ \sqrt{p} \end{pmatrix}$.

Alguns exemplos de funções *Kernel* mais estudadas na literatura e amplamente usadas em aprendizagem computacional para reconhecimento de padrões são dados na Tabela 2.1.

<i>Kernel</i>		Parâmetro
Linear:	$x^T z$	
Polinomial:	$K(x, z) = (x^T z + p)^d$	Grau $d \in \mathbb{N}$ e $p \in \mathbb{R}^+$
Gaussiano:	$K(x, z) = \exp\left(-\frac{\ x - z\ ^2}{2\sigma^2}\right)$	$\sigma \in \mathbb{R}^+$
Exponencial:	$\exp(\sigma x^T z)$	$\sigma \in \mathbb{R}^+$
Laplaciano:	$\exp(-\sigma \ x - z\)$	$\sigma \in \mathbb{R}^+$
Quadrático racional:	$\left(1 + \frac{\ x - z\ ^2}{\alpha\beta^2}\right)^{-\alpha}$	$\alpha, \beta \in \mathbb{R}^+$
Multi-quadrático:	$-\sqrt{\ x - z\ ^2 + \alpha^2}$	$\alpha \in \mathbb{R}^+$
Multi-quadrático inverso:	$\left(\sqrt{\ x - z\ ^2 + \alpha^2}\right)^{-1}$	$\alpha \in \mathbb{R}^+$
Distância inversa:	$1/(\ x - z\ + \sigma)$	$\sigma \in \mathbb{R}^+$
Distância inversa quadrática:	$1/(\ x - z\ ^2 + \sigma)$	$\sigma \in \mathbb{R}^+$
Delta:	$\delta(x, z) = \begin{cases} 1, & x = z \\ 0, & \text{cc} \end{cases}$	
Sigmoidal:	$K(x, z) = \tanh(k(x^T z) + \theta)$	k e $\theta < 0$

Tabela 2.1: Exemplos de funções *Kernel*.

Outros exemplos de funções *Kernel* podem ser encontrados no trabalho de Smola e Schölkopf [10].

O Truque do *Kernel* pode ser aplicado a outras técnicas de aprendizagem, desde que possa ser escrito em termos de produtos internos entre os dados de entrada, permitindo assim que o algoritmo trabalhe de forma eficiente em um espaço de dimensão elevada.

No Exemplo 2.42, ilustramos a questão de similaridade da função *Kernel* Gaussiana.

Exemplo 2.42 Considere os pontos de treinamento $X = \{x^1, x^2, x^3\}$, com saídas $y = \{1, 1, -1\}$, conforme Figura 2.14(a). Imagine que ao treinar tal conjunto com a função Kernel RBF, obtemos $\alpha^* = (1, 0.5, 0.7)^T$, $w^* = (1, 1, 0)^T$ e $b^* = -1/2$. Para o novo ponto \tilde{x} , temos que $K(\tilde{x}, x^1) \approx 1$ mas $K(\tilde{x}, x^2) \approx 0$ e $K(\tilde{x}, x^3) \approx 0$, veja Figura 2.14(b). Pela função de decisão (2.61) temos

$$F(\tilde{x}) = \text{sinal} \left(\sum_{i=1}^3 \alpha_i^* y_i K(x^i, \tilde{x}) + b^* \right) \approx \text{sinal} (1 - 0.5) = \text{sinal} (0.5),$$

logo, prediz que \tilde{x} pertence à classe positiva. Para o ponto \hat{x} , temos que $K(\hat{x}, x^3) \approx 1$ mas $K(\hat{x}, x^1) \approx 0$ e $K(\hat{x}, x^2) \approx 0$, veja Figura 2.14(c). Logo prediz que \hat{x} pertence à classe negativa, pois $F(\hat{x}) \approx \text{sinal} (-1)$.

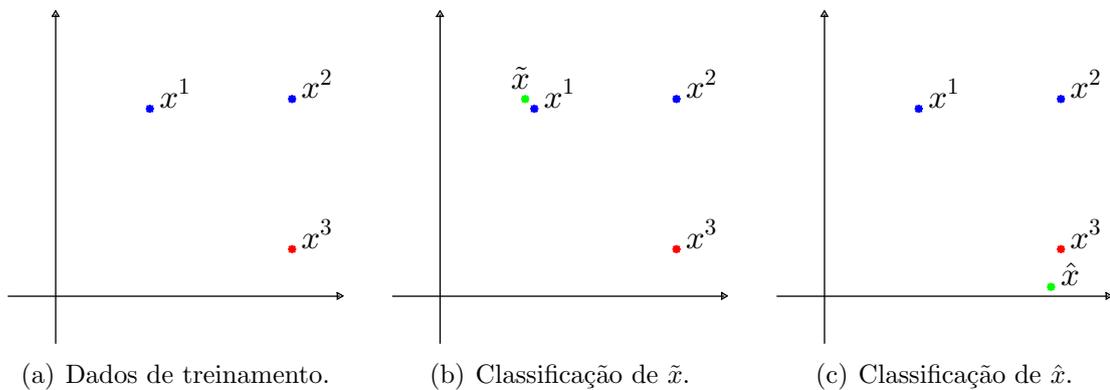


Figura 2.14: Similaridade entre os dados.

Capítulo 3

Experimentos numéricos

Este capítulo é dedicado a experimentos numéricos, com a finalidade de abordar pontos vistos na parte teórica tratada nos capítulos anteriores. Para grande parte das implementações deste trabalho, utilizamos o *software* Matlab em sua versão R2015a. Os testes foram realizados em um computador portátil com processador Intel Core™ i3-4010U com 3 MB de memória cache, com velocidade do clock de 1.7 GHz e com 4 GB de memória RAM, com sistema operacional Windows 8.1 Single com arquitetura 64 bits.

Usamos também o projeto Algencan, que foi desenvolvido em Fortran, mas que recorreremos à sua interface em Matlab, com sistema operacional Linux e Matlab em sua versão R2012a, utilizando para isso o Método do Lagrangiano Aumentado. Os realizadores deste projeto são os professores Ernesto Birgin da Universidade de São Paulo e José Mario Martínez da Universidade Estadual de Campinas. Agradecemos ao professor Francisco Nogueira Calmon Sobral, da Universidade Estadual de Maringá, que gentilmente nos auxiliou na instalação e utilização do Algencan.

O capítulo é dividido em duas seções. Na primeira, apresentamos alguns exemplos simples de dados, com o objetivo de visualizar graficamente tais dados e os hiperplanos separadores, além de possibilitar a análise dos resultados obtidos, juntamente com a teoria estudada no Capítulo 2. A Seção 3.2 é dedicada à verificação da eficiência de Máquinas de Vetores Suporte em um problema real. Para isto, realizamos alguns experimentos em classificação de caracteres. Nos testes usamos um banco de dados criado artificialmente.

Apresentamos no Apêndice A a implementação utilizada para resolver os problemas deste trabalho, usando a interface do Algencan e funções próprias do Matlab. Mostramos a instalação do Algencan, desde seu *download* até os principais passos para sua utilização.

3.1 Metodologia

A solução da SVM depende da seleção dos parâmetros da função *Kernel* utilizada e do parâmetro de regularização C . Não é possível saber antecipadamente qual função

Kernel é mais adequada, assim como seus parâmetros. Para maiores detalhes sobre funções *Kernel* podemos consultar [15, 16]. Assim, para os problemas apresentados, usamos o *Kernel* RBF, por ser um dos mais utilizados e com bons resultados em problemas reais, alguns exemplos podem ser vistos em [1, 7].

Para obter os melhores valores do parâmetro C , o qual determina um ponto de equilíbrio entre a maximização da margem e a minimização do erro de classificação, podemos usar uma grade de busca, escolhendo por exemplo, $C = 2^{-5}, 2^{-3}, \dots, 2^{15}$ [2, 27]. Em [35] foram utilizados $C = 10^i, i = 5, \dots, 12$ para regressão.

Os melhores parâmetros das funções *Kernel* podem ser encontrados por validação cruzada, que consiste em dividir o conjunto de dados em k subconjuntos. Sequencialmente, cada um desses subconjuntos é testado, usando o classificador treinado com os demais $(k - 1)$ subconjuntos [27].

A seguir, tratamos da SVM com margem rígida, restrito a poucas aplicações práticas, da SVM com margem flexível e da não linear, respectivamente. Para todas elas, consideramos o conjunto de treinamento

$$X = \{(x^1, y_1), \dots, (x^m, y_m), x^i \in \mathbb{R}^n, y_i \in \{-1, 1\}\},$$

para o problema de classificação binária.

Pela Observação 1.21, usamos minimização no restante do texto. Assim, resolvemos $-\min L(\alpha)$ ao invés de $\max L(\alpha)$, que facilita na implementação computacional.

3.1.1 Implementação SVM - Margem rígida

Conforme tratamos no Capítulo 2, o problema primal para a técnica SVM com margem rígida pode ser formulado por (2.3) e o dual por (2.10). Mostramos uma aplicação no Exemplo 3.1.

Exemplo 3.1 *Consideremos o conjunto de treinamento*

$$TR = \{(-5, -1), (-4, -6), (-4, 5), (-3, -2), (0, 1), (-2, -5), (2, -4), (3, 1), (4, 4), (5, -2)\}$$

com saídas $y_{tr} = \{1, 1, 1, 1, 1, -1, -1, -1, -1, -1\}$.

Resolvendo o problema (2.3), com o auxílio da função *quadprog* do *Matlab*, obtemos $w^* = (-1.4, 0.8)^T$ e $b^* = 0.2$, conseqüentemente o valor da função objetivo é $f(w^*, b^*) = \frac{1}{2} \|w^*\|^2 = 1.3$. Assim, o hiperplano separador é dado por $-1.4x_1 + 0.8x_2 = -0.2$. Conforme podemos acompanhar na Figura 3.1, os pontos que estão sobre os hiperplanos H_1 e H_2 , em tracejado, são $(-4, -6)$, $(0, 1)$ e $(-2, -5)$.

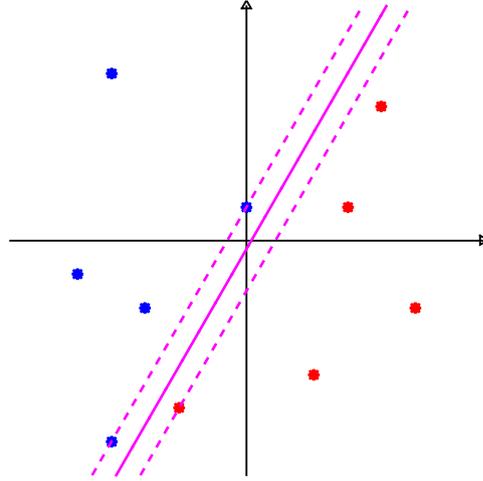


Figura 3.1: Hiperplano separador para um conjunto linearmente separável.

A seguir, resolvemos este problema, usando a formulação dual Lagrangiana,

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i,j=1}^{10} \alpha_i \alpha_j y_i y_j (x^i)^T x^j - \sum_{i=1}^{10} \alpha_i \\ \text{s.a.} \quad & \sum_{i=1}^{10} \alpha_i y_i = 0 \\ & -\alpha_i \leq 0 \quad i = 1, \dots, 10. \end{aligned}$$

As condições de KKT para tal problema, com $g_i(\alpha) = -\alpha_i, i = 1, \dots, 10$ e $h(\alpha) = \sum_{i=1}^{10} \alpha_i y_i$, são as seguintes

$$\begin{bmatrix} \sum_{i=1}^{10} \alpha_i^* y_1 y_i (x^1)^T x^i - 1 \\ \sum_{i=1}^{10} \alpha_i^* y_2 y_i (x^2)^T x^i - 1 \\ \vdots \\ \sum_{i=1}^{10} \alpha_i^* y_{10} y_i (x^{10})^T x^i - 1 \end{bmatrix} - \begin{bmatrix} \beta_1^* \\ \beta_2^* \\ \vdots \\ \beta_{10}^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (3.1a)$$

$$g_i(\alpha^*) = -\alpha_i^* \leq 0, \quad i = 1, \dots, 10 \quad (3.1b)$$

$$\beta_i^* \geq 0, \quad i = 1, \dots, 10 \quad (3.1c)$$

$$\beta_i^* g_i(\alpha^*) = 0, \quad i = 1, \dots, 10. \quad (3.1d)$$

$$h(\alpha^*) = \sum_{i=1}^{10} \alpha_i^* y_i = 0. \quad (3.1e)$$

Resolvendo o problema (2.10) com os dados deste problema e recorrendo ao quadprog, obtemos

$$\alpha^* = (0, 1, 0, 0, 0.3, 1.3, 0, 0, 0, 0)^T, \quad (3.2)$$

que contém uma quantidade considerável de componentes não nulas, logo não precisamos recorrer ao Algoritmo 2.1.

Note que tal solução satisfaz as condições de KKT descritas acima. De fato,

$$h(\alpha^*) = \sum_{i=1}^{10} \alpha_i^* y_i = \alpha_2^* y_2 + \alpha_5^* y_5 + \alpha_6^* y_6 = 1.1 + 1.0,3 - 1.1,3 = 0 \text{ e}$$

perceba que (3.1b) sai naturalmente. De (3.1a) obtemos $\beta_1^* = 5.2, \beta_2^* = 0, \beta_3^* = 8.6, \beta_4^* = 1.6, \beta_5^* = 0, \beta_6^* = 0, \beta_7^* = 7, \beta_8^* = 3.4, \beta_9^* = 1.4$ e $\beta_{10}^* = 6.4$, implicando que as relações (3.1c) e (3.1d) são satisfeitas. Assim, o valor da função objetivo dual é dado por $-f(\alpha^*) = \frac{1}{2} \sum_{i=1}^{10} \alpha_i^* \alpha_j^* y_i y_j (x^i)^T x^j - \sum_{i=1}^{10} \alpha_i^* = -(-1.3)$.

A partir da solução (3.2), encontramos o valor de w^* , considerando apenas α_2^*, α_5^* e α_6^* , pois são os únicos $\alpha_i^* \neq 0$. Ao resolver o problema, recorrendo ao quadprog e a interface do Algencan, mesmo tomando pontos iniciais distintos, não obtivemos outras soluções α^* com menor quantidade de $\alpha_i^* \neq 0$. Portanto, pelas definições 2.11 e 2.12 os vetores suporte para tal problema são os relacionados aos $\alpha_i^*, i = \{2, 5, 6\}$. Então por 2.11,

$$w^* = \sum_{i \in I} \alpha_i^* y_i x^i = \alpha_2^* y_2 x^2 + \alpha_5^* y_5 x^5 + \alpha_6^* y_6 x^6 = \begin{pmatrix} -1.4 \\ 0.8 \end{pmatrix}.$$

Calculamos para o Exemplo 3.1 diferentes formas de obter b^* . Como mostramos que a solução é única, verificamos se ocorre igualdade nos resultados.

Primeiro, usamos 2.12 e fixamos $i = 2$, para obter b^* ,

$$b^* = \frac{-y_2 (w^*)^T x^2 + 1}{y_2} = y_2 - \sum_{j \in I} \alpha_j^* y_j (x^j)^T x^2 = 1 - (-1.4 \ 0.8) \begin{pmatrix} -4 \\ -6 \end{pmatrix} = 0.2.$$

Um outro modo de obter b^* , pode ser tomando x^i da classe positiva e x^j da classe negativa, que estejam sobre H_1 ou H_2 e teremos

$$(w^*)^T x^i + b^* = 1 \text{ e} \tag{3.3}$$

$$(w^*)^T x^j + b^* = -1, \tag{3.4}$$

somando (3.4) e (3.3), depois isolando b^* obtemos

$$b^* = -\frac{1}{2} (w^*)^T (x^i + x^j). \tag{3.5}$$

Deste modo, tomando $x^2 \in X_1$ e $x^6 \in X_2$ temos que

$$b^* = -\frac{1}{2} (-1.4 \ 0.8) \left[\begin{pmatrix} -4 \\ -6 \end{pmatrix} + \begin{pmatrix} -2 \\ -5 \end{pmatrix} \right] = 0.2.$$

Por [15] encontramos uma terceira forma de obter o valor de b^*

$$b^* = -\frac{\max_{y_i=-1}(w^*)^T x^i + \min_{y_i=1}(w^*)^T x^i}{2}. \quad (3.6)$$

Portanto, calculando $\min_{y_i=1}(w^*)^T x^i = \min_{y_i=1}\{(w^*)^T x^2, (w^*)^T x^5\} = \min\{0.8, 0.8\} = 0.8$ e $\max_{y_i=-1}(w^*)^T x^i = \max_{y_i=-1}\{(w^*)^T x^6\} = \max_{y_i=-1}\{-1.2\} = -1.2$, obtemos

$$b^* = -\frac{0.8 - 1.2}{2} = 0.2.$$

Note que em [15] não são tomados arbitrariamente os dados $x^i \in X_1$ e $x^j \in X_2$, conforme procedemos no cálculo anterior. Entretanto, realizamos mais operações.

Por [6] temos mais uma maneira de obter b^* , com a justificativa de evitar erros de arredondamento, utilizando a média entre b^* calculado com cada vetor de suporte

$$b^* = \frac{1}{3} \sum_{i \in I} \left(y_i - \sum_{j=1}^{10} \alpha_j^* y_j (x^j)^T x^i \right), \quad (3.7)$$

com $I = \{i \mid \alpha_i^* > 0\}$, o que equivale a

$$\begin{aligned} b^* &= \frac{1}{3} \left[y_2 + y_5 + y_6 - \left(\sum_{j \in I} \alpha_j^* y_j (x^j)^T x^2 + \sum_{j \in I} \alpha_j^* y_j (x^j)^T x^5 + \sum_{j \in I} \alpha_j^* y_j (x^j)^T x^6 \right) \right] \\ &= \frac{1}{3} (1 - 0.4) = 0.2. \end{aligned}$$

Observe que podemos alterar o segundo somatório da fórmula de [6], já que os $\alpha_j \notin I$ são iguais a zero, reescrevendo obtemos

$$b^* = \frac{1}{|I|} \sum_{i \in I} \left(y_i - \sum_{j \in I} \alpha_j y_j (x^j)^T x^i \right),$$

conforme podemos consultar na página 330 de [3].

Para validação da função de decisão obtida, $F(x) = \text{sin}al((w^*)^T x + b^*)$, procedemos para a fase de teste, onde tomamos um conjunto, que já sabemos a priori sua classificação, e verificamos se $F(x)$ classifica corretamente tal conjunto. Tomamos como conjunto de teste,

$$TE = \{(-6, -1), (-6, 1), (-5, 2), (-5, 3), (-5, -3), (5, 4), (5, -3), (4, 1), (2, -5), (3, -4)\},$$

com saídas $y_{te} = \{1, 1, 1, 1, 1, -1, -1, -1, -1, -1\}$.

Para medir a acurácia, calculamos

$$\text{Acurácia} = \frac{V_p + V_n}{V_p + F_p + V_n + F_n},$$

que fornece a porcentagem de amostras positivas e negativas classificadas corretamente sobre todas as amostras. Consideramos V_p sendo o número de classificações verdadeiras positivas, ou seja, número de dados que a função de decisão diz pertencer à X_1 e realmente pertence, V_n é o número de classificações verdadeiras negativas, ou seja, número de dados que a função de decisão diz pertencer à X_2 e realmente pertence, F_p é o número de classificações falsas positivas, ou seja, que a função de decisão diz pertencer à X_1 mas que realmente pertence à X_2 e F_n é o número de classificações falsas negativas, ou seja, que a função de decisão diz não pertencer à X_2 mas que realmente pertence à X_1 .

Temos que V_p, V_n, F_p e F_n são obtidos pela matriz de confusão, que mostra o número de classificações reais contra as previstas para cada classe, sobre o conjunto de teste. O número de acertos, para cada classe, se localiza na diagonal principal e os demais elementos da matriz representam erros na classificação. Para uma função de decisão ideal, todos os elementos fora da diagonal são zero, já que não comete erros.

Classe	Predita X_1	Predita X_2
Verdadeira X_1	V_p	F_n
Verdadeira X_2	F_p	V_n

Tabela 3.1: Matriz de confusão para 2 Classes.

Calculando a acurácia para o conjunto de teste TE obtemos 100% de acerto, conforme podemos acompanhar pela Figura 3.2 e Tabela 3.2, onde as bolinhas abertas representam dados de teste.

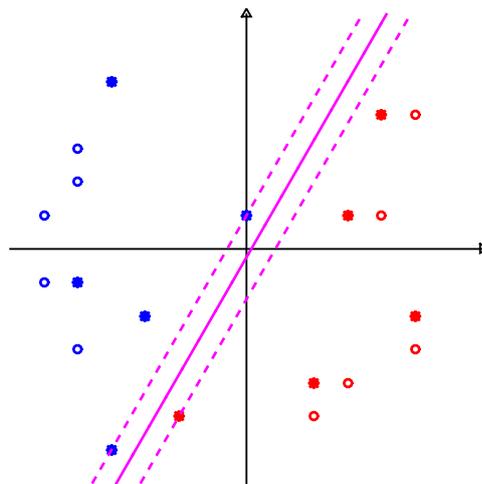


Figura 3.2: Hiperplano separador com conjunto de teste.

Classe	Predita X_1	Predita X_2
Verdadeira X_1	5	0
Verdadeira X_2	0	5

Tabela 3.2: Matriz de confusão para conjunto de teste.

Para utilizarmos o Algoritmo 2.1, recorreremos ao Algencan que forneceu soluções distintas ao alterarmos apenas o ponto inicial no programa. Ao contrário do quadprog, que forneceu sempre a mesma solução, independentemente do ponto inicial e do algoritmo escolhido.

3.1.2 Aplicação com Algencan

No Exemplo 2.10, recorrendo ao quadprog, obtemos apenas a solução α^* , mesmo tomando pontos iniciais distintos e alterando o algoritmo utilizado (*interior-point-convex (default)*, *trust-region-reflective* e *active-set*). Assim, usamos à interface do Algencan em Matlab, que tomando pontos iniciais distintos forneceu soluções duais distintas. Apresentamos a aplicação do Algoritmo 2.1 em tal exemplo.

Para o Algoritmo 2.1 tomamos sempre como ponto inicial um vetor randômico. Na primeira iteração, resolvemos o problema dual (2.10), obtendo uma solução com apenas a quarta componente nula. Assim, reduzimos o conjunto de treinamento para $TR = \{(0, 1), (1, 1), (2, 1), (1, -1), (2, -1)\}$. Na iteração seguinte, resolvemos o problema dual reduzido (2.14), obtendo uma solução com a primeira componente nula. Novamente, reduzimos o conjunto de treinamento para $TR = \{(1, 1), (2, 1), (1, -1), (2, -1)\}$. Por fim, na terceira e última iteração, resolvendo o dual reduzido (2.14), obtemos uma solução com a primeira e terceira componentes nulas. Portanto, encontramos uma quantidade reduzida de vetores suporte, que são os pontos $(2, 1)$ e $(2, -1)$. Na Tabela 3.3 mostramos os resultados obtidos ao longo das iterações.

Iteração	Ponto inicial aproximado	Solução aproximada
1	$(0.15, 0.26, 0.84, 0.25, 0.81, 0.24)^T$	$(0.18, 0.03, 0.29, 0, 0.39, 0.11)^T$
2	$(0.35, 0.83, 0.59, 0.55, 0.92)^T$	$(0, 0.22, 0.28, 0.22, 0.28)^T$
3	$(0.78, 0.93, 0.13, 0.57)^T$	$(0, 0.5, 0, 0.5)^T$

Tabela 3.3: Soluções obtidas pelo Algoritmo 2.1.

Para classificar novos dados, realizamos menos cálculos na função de decisão

$$F(x) = \text{signal}((w^*)^T x + b^*) = \text{signal}\left(\sum_{i \in \{3,6\}} \alpha_i^* y_i x^i - y_2 - \sum_{j \in \{3,6\}} \alpha_j^* y_j x^2\right).$$

Considere agora um problema com mais dados e que nem todos estejam sobre os hiperplanos H_1 e H_2 .

Exemplo 3.2 Seja $TR = \{(-3, 1), (-2, 2), (-1, 1), (0, 2), (1, 1), (2, 2), (3, 1), (4, 2), (5, 1), (6, -1), (5, -2), (4, -1), (3, -2), (2, -1), (1, -2), (-1, -2), (-2, -1), (-3, -2)\}$ o conjunto de treinamento e $ytr = \{1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1\}$ suas saídas, conforme Figura 3.3.

Na primeira iteração do Algoritmo 2.1, eliminamos todos os dados que não estão sobre os hiperplanos H_1 e H_2 . Tomamos para a próxima iteração apenas os dados da Figura 3.4(a). Na segunda iteração, eliminamos os dados associados ao $\alpha_i^* = 0$, sobrando os dados da Figura 3.4(b). Na terceira iteração, novamente redefinimos nosso conjunto de treinamento para os dados da Figura 3.4(c).

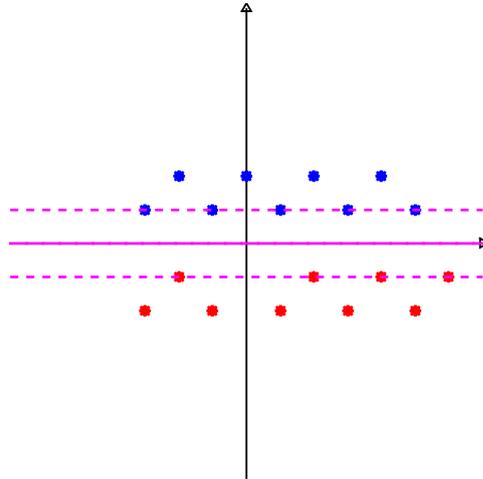


Figura 3.3: Dados linearmente separáveis, para aplicação do Algoritmo 2.1.

Na quarta iteração, eliminamos o primeiro dado deste conjunto de treinamento, sobrando apenas os dados $(5, 1), (6, -1)$ e $(2, -1)$, conforme Figura 3.4(d). Assim, paramos as iterações pois não é possível obter um α^* menor. Isto ocorre, pois não encontramos nenhum segmento perpendicular ao hiperplano que ligue o dado restante da classe positiva a algum dos outros dois dados da classe negativa. De fato, se fosse possível obter tal α^* iríamos obter outros hiperplanos, conforme Figura 3.5, contrariando o Teorema 2.9. Na Tabela 3.4 mostramos as soluções obtidas pelo Algoritmo 2.1.

Iteração	Solução aproximada
1	$(0.07, 0, 0.08, 0, 0.09, 0, 0.1, 0, 0.15, 0.09, 0, 0.1, 0, 0.11, 0, 0, 0.17, 0)^T$
2	$(0, 0.24, 0, 0, 0.26, 0.2, 0, 0.11, 0.18)^T$
3	$(0.21, 0.28, 0.05, 0.45, 0)^T$
4	$(0, 0.5, 0.37, 0.12)^T$

Tabela 3.4: Soluções obtidas pelo Algoritmo 2.1.

Assim, uma possibilidade para critério de parada do Algoritmo 2.1 seria iterar até m ser menor que $n + 1$. Não sabemos se tal critério poderia demorar a ocorrer para problemas mais gerais, mas é um bom limitante. Pensamos também em parar o algoritmo quando para um certo número de iterações a solução α^* fornecida não gerar componentes

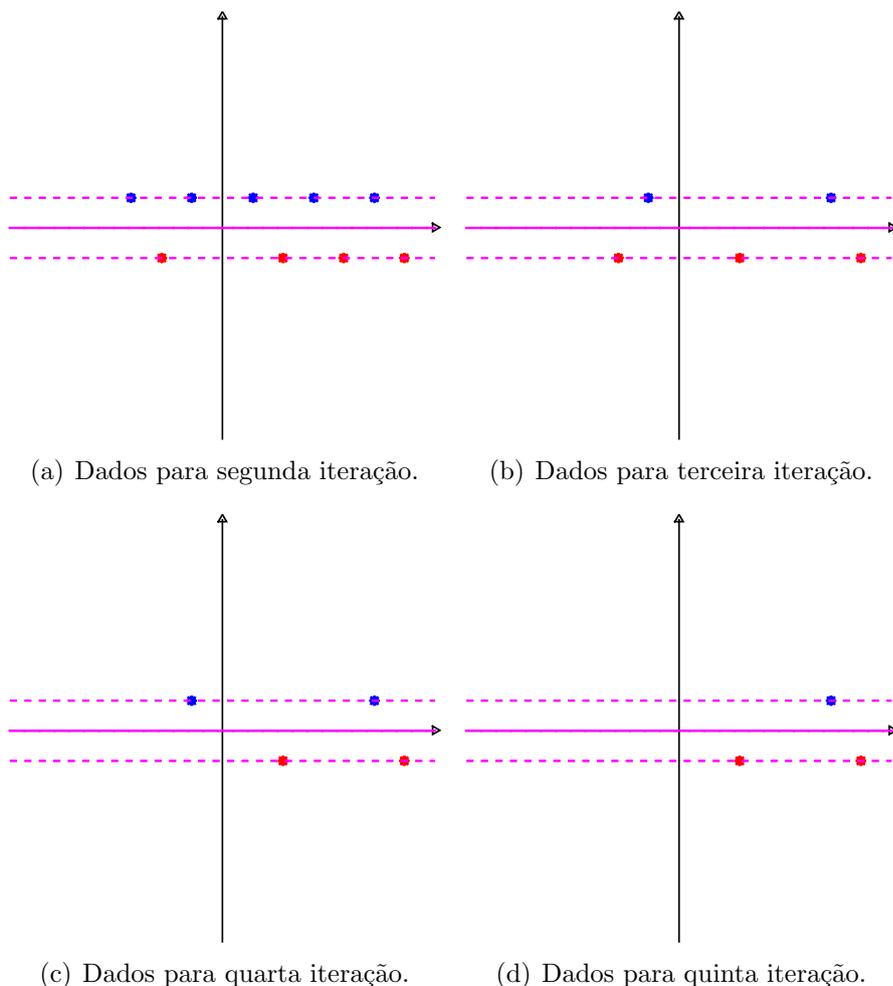


Figura 3.4: Eliminação de dados pelo Algoritmo 2.1.

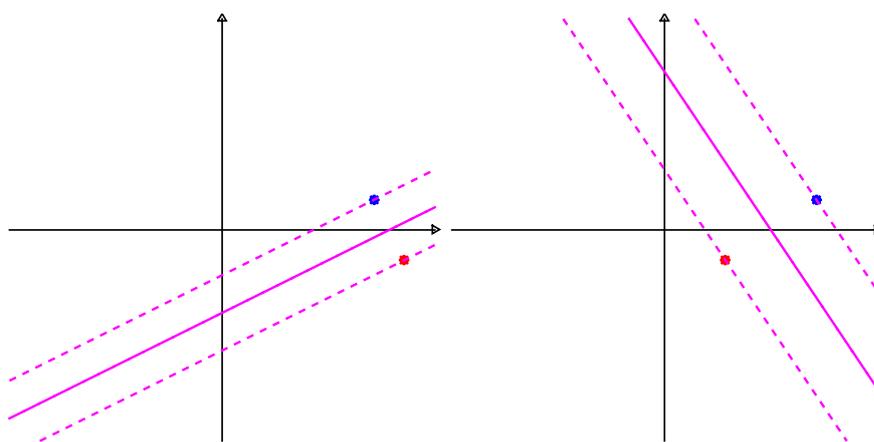


Figura 3.5: Hiperplanos para somente dois dados, não perpendiculares.

nulas, isso pode ser ariscado já que podemos iterar e obter só soluções com α_i^* s não nulos e perdermos uma solução com grande quantidade de α_i^* s nulos. Enfim, isso deve ser algo que precisa ser considerado dependendo do problema, pois a procura de um α^* com dimensão pequena pode ter um custo computacional maior que para a função de decisão usando α^* com dimensão maior.

3.1.3 Implementação SVM - com margem flexível

Agora, consideramos que o conjunto de dados não seja linearmente separável. Conforme tratamos no Capítulo 2, para tal conjunto podemos resolver o problema (2.17).

Exemplo 3.3 *Considerando novamente o conjunto TR do exemplo anterior, mas agora com $(-5, -1)$ pertencendo ao conjunto X_2 e $(2, -4)$ ao conjunto X_1 . Assim, TR deixa de ser linearmente separável, então não podemos resolver o problema (2.3), pois obtemos $exitflag=-2$, recorrendo ao quadprog, ou seja, o resultado não é viável já que o conjunto é vazio. Deste modo, usamos regularização, ou seja, resolvemos o problema,*

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{10} \xi_i & w \in \mathbb{R}^2, b \in \mathbb{R}, \xi \in \mathbb{R}^{10} \\ \text{s.a} \quad & y_i(w^T x^i + b) \geq 1 - \xi_i, \quad i = 1, \dots, 10, \\ & \xi_i \geq 0, \quad i = 1, \dots, 10. \end{aligned} \tag{3.8}$$

Para a escolha do parâmetro de regularização C , passamos à fase de teste, considerando o mesmo conjunto de teste do Exemplo 3.1. Na Tabela 3.5 apresentamos os resultados, alterando os valores de C .

C	$fval$	$exitflag$	w^*	b^*
10^{-3}	0.0096	1	$\begin{pmatrix} -0.0280 \\ -0.0030 \end{pmatrix}$	-0.1299
10^{-2}	0.0638	1	$\begin{pmatrix} -0.1978 \\ 0.0218 \end{pmatrix}$	0.0327
10^{-1}	0.2961	1	$\begin{pmatrix} -0.3810 \\ 0.0952 \end{pmatrix}$	0.0476
1	1.2308	1	$\begin{pmatrix} -1.0769 \\ 0.6154 \end{pmatrix}$	0.3846
$10^i, i = 1, \dots, 5$	1.3	1	$\begin{pmatrix} -1.4 \\ 0.8 \end{pmatrix}$	0.2

Tabela 3.5: Resultados para diferentes valores do parâmetro C .

A saída 1 do $exitflag$ diz que a primeira condição de otimalidade foi satisfeita. Notamos que, à medida que o parâmetro C aumenta, o valor da função objetivo cresce. Além disso, para os parâmetros $C = 10^i, i = 1, \dots, 5$, obtemos a mesma solução que o exemplo anterior, mesmo alterando o ponto inicial no quadprog. Isto sugere que a solução ótima é única a partir de um C próximo de 10. Na Figura 3.6 apresentamos os hiperplanos separadores obtidos para $C = 10^{-3}$, $C = 10^{-1}$, $C = 1$ e $C = 10$ e na Tabela 3.6 a acurácia para diferentes C . Note que, quanto maior o parâmetro C , menor a margem de separação.

C	Acurácia
10^{-3}	50%
$10^i, i = -2, \dots, 5$	100%

Tabela 3.6: Acurácia para diferentes valores do parâmetro C .

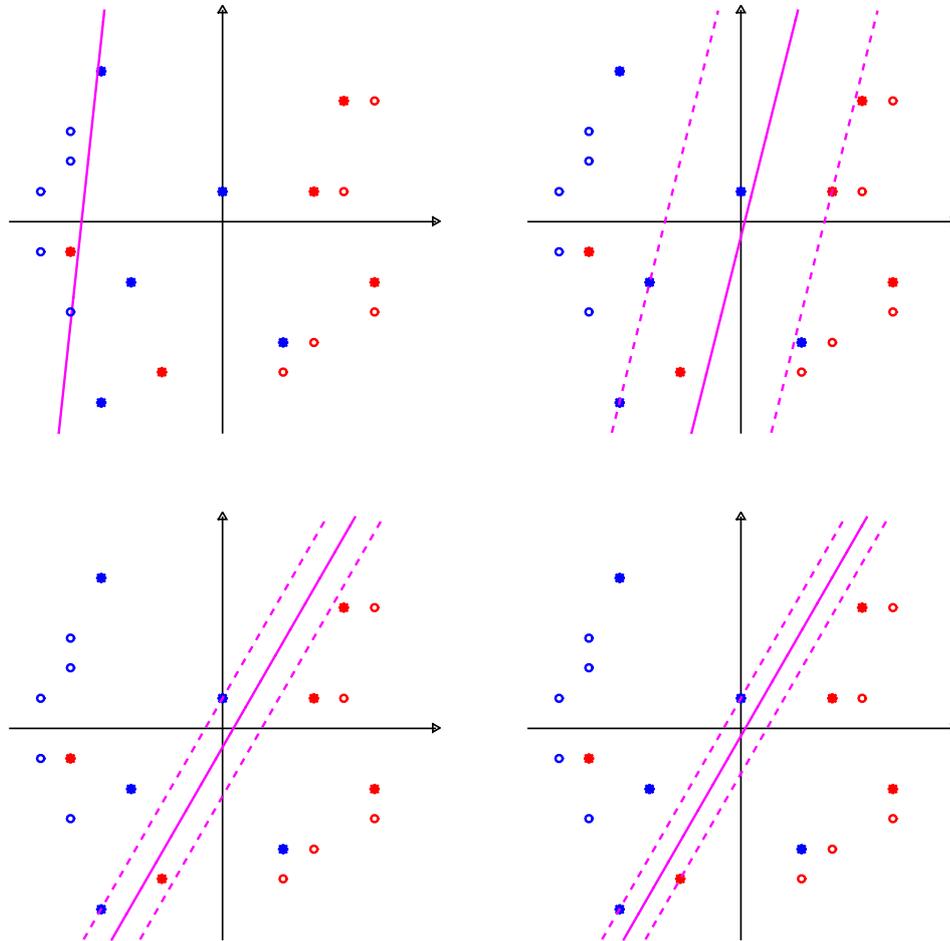


Figura 3.6: Dados de teste representados pelos círculos abertos. Hiperplanos separadores obtidos com parâmetros $C = 10^{-3}$, $C = 10^{-1}$, $C = 1$ e $C = 10$, respectivamente.

3.1.4 Implementação SVM - não linear

Conforme tratamos no Capítulo 2, para resolvermos o primal precisamos conhecer a função ϕ , que é um mapeamento do espaço de entrada no espaço de características, isto é, a cada $x^i \in \mathbb{R}^n$ associamos $\phi(x^i) = z^i \in \mathbb{R}^N$, com $N > n$, com função *Kernel* $K(x^i, x^j) = \phi(x^i)^T \phi(x^j)$, descrito no problema (2.53). Como vimos, para o problema dual não é necessário conhecer a função ϕ , precisando apenas do produto interno $K(x^i, x^j)$. Para a SVM não linear com regularização, recorrendo ao dual, precisamos resolver o problema (2.55).

Exemplo 3.4 Consideramos agora o seguinte conjunto de treinamento $TR = \{(-5, -1), (-4, -6), (-4, 5), (-3, -2), (-2, -5), (0, 1), (2, -4), (3, 1), (4, 4), (5, -2), (1, 5)\}$ e as seguintes saídas $y_{tr} = \{1, 1, 1, 1, -1, 1, -1, -1, -1, -1, -1\}$. Note pela Figura 3.7 que não conseguimos obter um classificador linear no espaço de entrada.

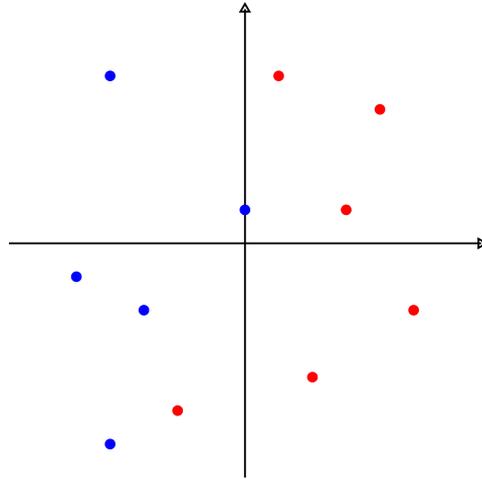


Figura 3.7: Conjunto de dados não linearmente separável.

Deste modo, tomamos as funções de mapeamento a seguir,

$$\phi_1(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}, \phi_2(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} x_1^2 - x_2^2 \\ 2x_1x_2 \\ x_1^2 + x_2^2 \end{pmatrix},$$

$$\phi_3(x) = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{pmatrix} \text{ e } \phi_4(x) = \begin{pmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix},$$

para tentar obter uma classificação linear dos dados.

Assim, substituímos as funções de mapeamento acima no problema (2.53), obtendo os resultados da Tabela 3.7, em que as saídas -2 e 1 do exitflag são resultado não viável e primeira condição de otimalidade satisfeita, respectivamente.

Vemos que, das funções mapeamento apresentadas, apenas a função ϕ_4 resolve o problema (2.53) para o conjunto de treinamento dado. Deste modo, nem toda função de mapeamento possibilita obter uma separação linear dos dados, conforme podemos acompanhar na Figura 3.8. Por razão da dimensão de ϕ_3 e ϕ_4 , não representamos as superfícies que tais mapeamentos descrevem.

Deste modo, considerando o mapeamento ϕ_4 , obtemos o hiperplano separador $H(z) = (w^*)^T z + b^* = 0$, com $z = \phi_4(x) \in \mathbb{R}^6$, logo $H(z) = w_1^* + w_2^*\sqrt{2}x_1 + w_3^*\sqrt{2}x_2 + w_4^*x_1^2 + w_5^*\sqrt{2}x_1x_2 + w_6^*x_2^2 + b^* = 0$. Portanto, obtemos a seguinte função

$$F(x) = 0.3989 - 0.6188\sqrt{2}x_1 + 0.2326\sqrt{2}x_2 + 0.0315x_1^2 + 0.0806\sqrt{2}x_1x_2 - 0.1267x_2^2 + 0.3989.$$

A superfície separadora no espaço de entrada, é obtida pela projeção da inter-

Função	Tempo em segundos	$fval$	$exitflag$	w^*	b^*
ϕ_1	2.5	0.3846	-2	$\begin{pmatrix} -0.0819 \\ -0.0409 \\ -0.0066 \end{pmatrix}$	0.8722
ϕ_2	0.1	0.3846	-2	$\begin{pmatrix} -0.0533 \\ -0.0409 \\ -0.0626 \end{pmatrix}$	0.8722
ϕ_3	0.4	0.3846	-2	$\begin{pmatrix} -0.0819 \\ -0.0289 \\ -0.0289 \\ -0.0066 \end{pmatrix}$	0.8722
ϕ_4	0.8	0.3894	1	$\begin{pmatrix} 0.3989 \\ -0.6188 \\ 0.2326 \\ 0.0315 \\ 0.0806 \\ -0.1267 \end{pmatrix}$	0.3989

Tabela 3.7: Resultados para mapeamentos distintos.

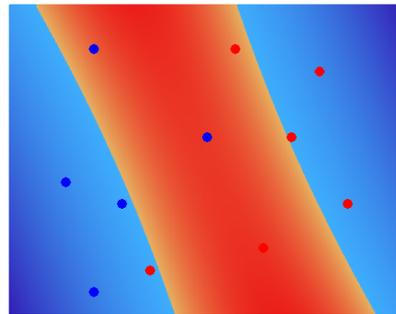
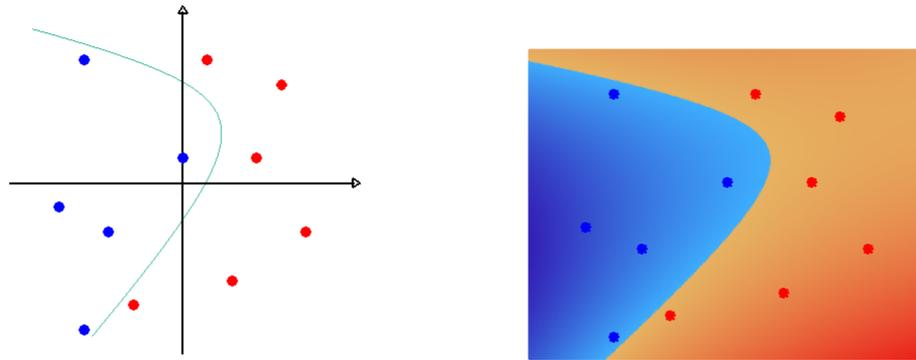


Figura 3.8: Região em azul: classe positiva. Região em vermelho: classe negativa. Superfície separadora no espaço de entrada para o mapeamento ϕ_1 .

secção do hiperplano separador no espaço de características com a superfície formada pela função de mapeamento. Neste caso, a superfície separadora obtida é uma hipérbole $F(x) = 0$, conforme Figura 3.9(a).

Nem sempre conseguimos expressar explicitamente a superfície separadora no espaço de entrada. Deste modo, analisamos o sinal da função $F(x)$. Veja Figura 3.9(b), em que as regiões em vermelho e azul, mais escuras, representam dados longes da superfície separadora.

Calculando $F(x)$ para todo x do conjunto de treinamento, podemos analisar suas posições em relação ao hiperplano $H(z)$. Por arredondamento, temos que $F(x^2) = 1, F(x^3) = 1, F(x^5) = 1, F(x^6) = -1, F(x^8) = -1$ e $F(x^{11}) = -1$, logo estão sobre os hiperplanos $H_1(z) = (w^*)^T z + b^* = -1$ e $H_2(z) = (w^*)^T z + b^* = 1$. Deste modo, estes são vetores suporte, segundo a Definição 2.11. Já para verificar a Definição 2.12, precisamos resolver o dual do problema.



(a) Superfície separadora no espaço de entrada.

(b) Regiões de classificação.

Figura 3.9: Hipérbole $F(x) = 0$.

Para verificar a validade do modelo, vamos à fase de teste e usamos novamente o conjunto de teste do Exemplo 3.1. A acurácia obtida foi de 100%, onde analisamos o sinal de $F(x)$ para cada x do conjunto de teste. Assim, não precisamos usar regularização.

No que segue, apresentamos a técnica SVM em um problema prático, que é a classificação de caracteres.

3.2 Classificação de caracteres

O objetivo foi encontrar classificadores por meio da SVM, treinados para identificar caracteres. Primeiramente, falamos de processamento de imagem, já que o banco de dados que utilizamos é composto por imagens.

Uma imagem é composta de um número finito de elementos tendo cada elemento um valor e uma localização particular, veja Figura 3.10. Cada elemento da imagem é chamado de pixel. O processamento de imagem é qualquer forma de processamento de dados, onde as entradas e as saídas são imagens, por exemplo, fotografias. É utilizado nas técnicas de AM, fornecendo as características da imagem para sua interpretação.



Figura 3.10: Exemplo de uma imagem digital e intensidades de alguns dos seus pixels.

O processamento de imagens pode ser dividido em três partes: o pré-processamento, em que é feita a correção e suavização da imagem, aumento de contraste e remoção de ruídos, o realce, em que é realizado a segmentação, partição da imagem em regiões ou objetos e reconhecimento individual, para melhorar a qualidade da imagem e por fim, a classificação que interessa, para poder atribuir classes através das características da imagem. O assunto vai muito além do que descrevemos neste texto, mas procuramos deixar claro cada etapa no problema de classificação de caracteres tratado nesta seção.

Existem inúmeras aplicações, como análise de recursos naturais e meteorologia por meio de imagens de satélites, análise de imagens médicas por ultra-som, radiação nuclear ou técnicas de tomografia computadorizada, transmissão digital de sinais de televisão, aplicações em automação industrial envolvendo o uso de sensores visuais em robôs.

Realizamos alguns testes numéricos para a classificação de caracteres de *A* a *Z*. Utilizamos um banco de dados, criado artificialmente, composto por 2574 imagens, sendo 99 de cada letra. Para construção das imagens, foram utilizadas 99 fontes do Excel, com tamanho 72 e cada cédula com 74.25 de altura e 13.43 de largura, depois foram transformadas em imagem no Paint, redimensionando em 100×100 pixels, com extensão png, conforme podemos acompanhar na Figura 3.11. Esse processo de redimensionamento da imagem pode ser realizado também com o uso do Matlab, mostramos como proceder no decorrer do texto.

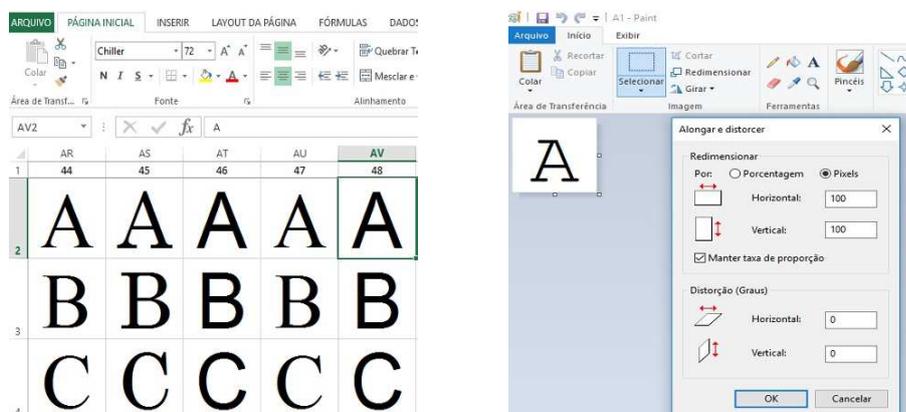


Figura 3.11: Criação dos caracteres e da imagem, respectivamente.

Manualmente, podemos classificar um conjunto pequeno de letras, por exemplo, dizer quais são *A* e quais não. Entretanto, para um conjunto maior de dados torna-se difícil o processo manual, sendo necessário um processamento automático da imagem para coletar os atributos. Para obter este processamento, recorreremos à técnica SVM.

Assim, precisamos encontrar uma representação numérica dos dados das imagens e transformá-los em vetores. Para este fim, recorreremos ao Matlab.

A seguir, descrevemos os principais comandos utilizados no Matlab para o processamento das imagens em nosso problema de classificação de caracteres. Dada a imagem

A1.png, que representa uma letra A, usamos o comando

```
» imshow('A1.png');
```

para visualizar a imagem e o comando

```
» rgb2gray(imread('A1.png'));
```

para fazer a leitura da imagem, que retorna uma matriz $A1 \in \mathbb{R}^{100 \times 100}$, do tipo uint8, inteiro de 8 bits sem sinal. Cada elemento $A1_{i,j}$ representa a tonalidade de cinza, que varia de 0 (cor preta) a 255 (cor branca), do pixel $\{i, j\}$ da imagem, conforme Figura 3.12. Em se tratando de uma imagem colorida, no sistema RGB (*red, green and blue*), cada pixel é especificado por 3 valores. A cor de cada pixel é assim determinada pela combinação de vermelho, verde e azul armazenada em cada pixel. Por exemplo, dada a imagem ararargb.jpeg, veja Figura 3.13, ao usarmos o comando

```
» [R,G,B]=imread('ararargb.jpeg');
```

obtemos uma matriz $m \times n \times 3$, com $R = (:, :, 1)$ componente vermelha, $G = (:, :, 2)$ componente verde e $B = (:, :, 3)$ componente azul.

```
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 58 0 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 0 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 0 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 0 144 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 182 0 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 102 0 255 0 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 219 255 0 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 255 255 0 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 255 255 0 219 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 255 255 0 0 255 255 255 255 255
255 255 255 255 255 255 255 182 0 0 0 0 0 255 255 255 255 255 255 255
255 255 255 255 255 255 255 0 0 0 0 0 0 255 255 255 255 255 255 255
255 255 255 255 255 255 255 0 255 255 255 255 0 255 255 255 255 255 255 255
255 255 255 255 255 255 255 0 255 255 255 255 0 255 255 255 255 255 255 255
255 255 255 255 255 255 255 0 255 255 255 255 0 0 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
```

Figura 3.12: Conversão da imagem em matriz.

Podemos fazer o processo inverso também, usando o comando `image(C)`, que mostra uma matriz C como uma imagem. Para alterar o tamanho de uma imagem, usamos o comando `imresize(A,M,'method')`, que retorna uma matriz M vezes maior (ou menor) que a imagem A , M pode ser também as novas dimensões: $M = [m, n]$. Os métodos usados podem ser: `nearest` = vizinho mais próximo, `bilinear` = interpolação bilinear ou `bicubic` = interpolação bicúbica.

Finalmente, após obtida a matriz $A1$, usamos o comando $A1 = A1(:)$ para con-



(a) Imagem colorida



(b) Tonalidade de vermelho



(c) Tonalidade de verde



(d) Tonalidade de azul

Figura 3.13: Imagem RGB.

catenar as colunas de cada imagem, obtendo o vetor que é usado na técnica SVM

$$A1 = \begin{bmatrix} A1_{\{1,1\}} \\ \vdots \\ A1_{\{100,1\}} \\ \vdots \\ A1_{\{1,100\}} \\ \vdots \\ A1_{\{100,100\}} \end{bmatrix} .$$

Em alguns problemas, o número de exemplos do treinamento e a alta dimensionalidade dos dados de entrada podem degradar seu desempenho computacional, tornando necessário tratar a dimensão e ou a grandeza dos dados numéricos, aplicando normalização e redução de dimensionalidade.

Normalização

Consiste em atribuir uma escala aos dados, por exemplo, o valor dos dados tomados apenas no intervalo $[-1, +1]$ [27]. Esse processo deve ser feito quando os dados estão com grandezas muito diferentes, o que pode tornar a matriz presente no modelo mal condicionada, possibilitando erros.

No problema proposto, classificação de caracteres, não usamos normalização, mas realizamos uma binarização do vetor $A1$ descrito acima, tomando limiar igual a 180 e

realizando o comando $A1 = A1 \geq \text{limiar}$. Assim se $A1(i) \geq 180$ obtemos $A1(i) = 1$ e para $A1(j) < 180$ obtemos $A1(j) = 0$.

Redução de dimensionalidade

Para a redução da dimensão dos dados originais, podemos usar a Análise de Componentes Principais (PCA, do Inglês *Principal Component Analysis*). Seus principais objetivos são: extrair informações redundantes e facilitar na interpretação dos resultados [3]. Útil para contornar o problema de dimensão elevada, que acarreta em um tempo computacional muito alto.

A PCA busca obter maior representatividade com respeito à matriz de variância covariância e os autovalores obtidos da mesma. Se o autovalor for grande, significa maior relevância na obtenção da solução. A matriz de variância covariância é quadrada, os elementos da diagonal contêm as variâncias das variáveis e os de fora da diagonal contêm as covariâncias entre todos os pares possíveis de variáveis. Por exemplo, dada uma matriz de variância covariância para três variáveis, X, Y e Z, as variâncias são exibidas ao longo da diagonal. A **variância** de X, Y e Z é 2,0, 3,4 e 0,82, respectivamente. A **covariância** entre X e Y é -0,86, conforme Tabela 3.8.

	X	Y	Z
X	2,0	-0,86	-0,15
Y	-0,86	3,4	0,48
Z	-0,15	0,48	0,82

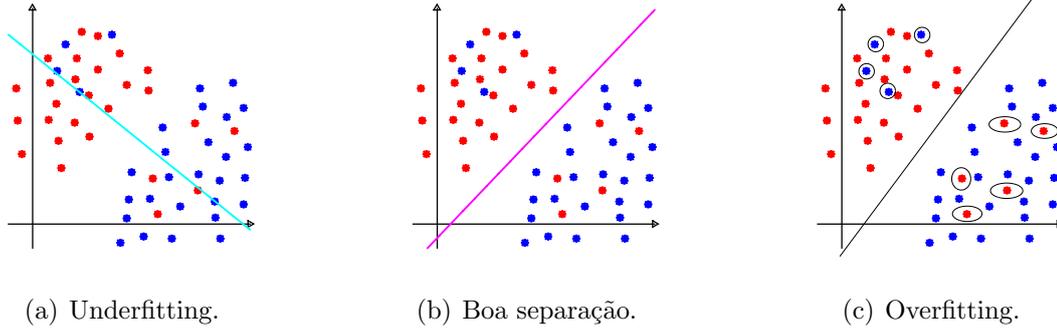
Tabela 3.8: Matriz de variância covariância.

A matriz de variância covariância é simétrica porque a covariância entre X e Y é a mesma covariância entre Y e X, onde $var(Y) = E((Y - E(Y))^2)$ e $cov(X, Y) = E(XY) - E(X)E(Y)$.

Não se sabe o número de variáveis ideais para cada problema, então, na prática, são eliminadas as variáveis cujos autovalores são menores que certa fração [3].

Se o número dos dados for pequeno em relação à sua dimensão, pode ocorrer um ajuste bom dos dados disponíveis, mas não para dados futuros da aplicação. Além disso, para ter sucesso na classificação, os dados de treinamento não devem ser um caso particular de alguns dados e precisam representar bem suas características. Pode ocorrer que o problema não fique generalizado na aprendizagem, ocorrendo um *overfitting*, veja Figura 3.14(c), ou *underfitting*, veja Figura 3.14(a), ou seja, que exagerou ou não houve um ajuste do modelo [15]. Na Figura 3.14(b) apresentamos uma possível classificação que foge destes casos.

Nosso problema de classificação de caracteres, apresenta dados em \mathbb{R}^{10000} . Primeiramente, não pensamos em usar a técnica PCA para contornar este problema de dimensão, mas sim o dual, pois a dimensão da solução α^* é igual a quantidade de dados



de treinamento 2574, que é bem menor. Entretanto, por questão de tempo computacional, reduzimos as matrizes das imagens, usando o comando

```
>>> im=imresize(A1,[20,20],‘nearest’);
```

em que, o tamanho da imagem A1 é reduzido para 20 por 20, pelo método nearest.

3.2.1 Critérios de classificação

A SVM foi originalmente concebida para ser aplicável diretamente somente para problemas binários (duas classes). Entretanto, a maior parte dos problemas reais envolvem múltiplas classes, por exemplo, o nosso problema requer 26 saídas. Para utilizarmos a SVM, é necessário fazer uma redução para diversos problemas binários. Para isso, podemos usar duas estratégias:

1. Estratégia *one-against-rest*: constrói classificadores binários que distinguem entre uma das classes e as demais. Assim, para n classes são construídos n classificadores. A classificação de novos dados é feita usando a estratégia o vencedor ganha, ou seja, o classificador com a maior saída atribui a classe [3].
2. Estratégia *one-against-one*: constrói um classificador para cada par de classes. Para n classes são construídos $\frac{n(n-1)}{2}$ classificadores. A classificação é feita por uma estratégia de votação maioria vence, onde cada classificador atribui uma das duas classes para o dado. A classe mais votada determina a classificação do dado [3].

Exemplo 3.5 *Suponha que tenhamos 4 classes A, B, C e D. Para a estratégia one-against-one, obtemos $\frac{4(4-1)}{2} = 6$ classificadores, os quais classificam em*

A ou B, A ou C, A ou D, B ou C, B ou D e C ou D.

Assim, para um novo dado, faremos a votação, é A ou B, por exemplo deu A, aí verificamos se é A ou C, por exemplo deu C e assim sucessivamente, se a classe mais votada foi A, então o dado pertence à classe das letras A.

Para a estratégia one-against-rest, teremos quatro classificadores para

A ou {B, C, D}, B ou {A, C, D}, C ou {A, B, D} e D ou {A, B, C}.

Assim, para um novo dado x , escolhemos a saída com valor máximo entre os quatro classificadores,

$$\bar{F}(x) = \arg \max_{1 \leq i \leq 4} ((w_i^*)^T x + b_i). \quad (3.9)$$

Infelizmente, essa abordagem heurística sofre com o problema de que os diferentes classificadores foram treinados em diferentes tarefas. Além disso, não há garantia de que o valor em (3.9) para diferentes classificadores terão escalas apropriadas [3].

A princípio iríamos usar a estratégia *one-against-rest*, já que a estratégia *one-against-one*, gera um tamanho grande de classificadores e a avaliação do resultado pode ser lenta. Entretanto, estávamos levando cerca de 30 minutos para obter cada classificador e em contrapartida 30 segundos para cada classificador na estratégia *one-against-one*. Além disso, tentamos obter funções de decisão, para o problema de classificação de caracteres, resolvendo os problemas (2.3) e (2.17), recorrendo a estratégia *one-against-rest*. Pois queríamos analisar se o conjunto de dados era linearmente separável, caso não fosse, verificaríamos qual o melhor parâmetro de regularização que reduzia erros na classificação durante a fase de teste. Entretanto, não conseguimos obter nenhum resultado, recorrendo ao Matlab, acreditamos ser o tamanho da matriz que armazena os dados.

Segundo Chang e Lin [12] temos que, embora na estratégia *one-against-one* muitos classificadores sejam treinados, cada classificador recebe menos dados de treinamento (dados de duas classes apenas) e o tempo de treinamento total usualmente é inferior ao tempo dispendido pela estratégia *one-against-rest*. Por isso, dentre tais estratégias a mais utilizada é *one-against-one*. Assim, obtemos $\frac{26(26-1)}{2} = 325$ funções de decisão, uma que classifica em A ou B , outra que classifica em \bar{A} ou C e assim sucessivamente.

Levamos em conta as diferentes combinações do conjunto de treinamento e consequentemente do conjunto de teste, além do parâmetro de regularização C e do parâmetro da função *Kernel*. Portanto, fizemos um banco de 30 problemas, dentre as $\frac{99!}{80!(99-80)!}$ combinações possíveis, composto de 80 dados do conjunto de treinamento, para obter nossos classificadores. Selecionamos de forma aleatória os conjuntos de treinamento, para tornar o problema mais próximo de uma situação real.

Conforme vimos, para os problemas primais (2.53) e (2.54), precisamos ter conhecimento da função ϕ . Como estamos considerando o *Kernel* RBF e não temos conhecimento qual é a função ϕ , tal que $\phi(\hat{x})^T \phi(\tilde{x}) = \exp\left(-\frac{\|\hat{x} - \tilde{x}\|^2}{2\sigma^2}\right)$, resolvemos diretamente o problema dual (2.55). Além disso, o uso de representações duais em SVM permite trabalhar com dados em dimensões elevadas.

3.2.2 Escolha dos parâmetros

Ao resolvermos os 30 problemas, precisamos de alguma maneira comparar, ao alterar os parâmetros, as soluções obtidas. Escolhemos aqueles que resultam em um classificador com maior acurácia.

Utilizamos as escolhas do parâmetro de regularização $C = 10^j, j = -3, -2, -1, 0, 1, 2, 3$, fixado o parâmetro $\sigma = 10^i, i = 1, 2$ da função *kernel* RBF, pois foram os que forneceram menores erros. Consideramos 7 parâmetros de regularização C que queremos comparar e 30 combinações de conjunto de treinamento, que pretendemos obter o melhor classificador. Para todos os parâmetros foram obtidos classificadores para todos os conjuntos de treinamento. Utilizamos como medida de comparação o erro, que é igual um menos a acurácia.

Para o classificador de A e B , fixado o parâmetro $\sigma = 10$, o parâmetro de regularização que forneceu menores erros, para as diferentes escolhas dos conjuntos de treinamento, foi $C = 10^3$. Entretanto, para os parâmetros $C = 10^i, i = -3, -2$ temos o mesmo comportamento e maiores erros, para todas as 30 escolhas de conjunto de treinamento. Para o parâmetro $\sigma = 10^2$, temos uma competição entre os parâmetros $C = 10^2$ e $C = 10^3$, para obter os menores erros.

Já para o classificador de B e C , fixado o parâmetro $\sigma = 10$, os parâmetros $C = 10^i, i = 0, 3$ mostram-se bem competitivos. Porém para os parâmetros $C = 10^i, i = -3, -1$ temos o mesmo comportamento e maiores erros. Para o parâmetro $\sigma = 10^2$, temos uma competição boa entre os parâmetros $C = 10^i, i = 1, 2, 3$, mostrando que para $C > 10$ temos maiores acurácias.

Notamos que a escolha tanto do parâmetro de regularização C quanto o da função *Kernel* σ sensibiliza muito a taxa de acerto. Para $\sigma = 10^j, j = -3, -2, -1, 0$ obtemos que o parâmetro $C = 10^3$ fornece menores erros e os demais parâmetros C apresentam resultados competitivos.

Pelos testes realizados, conseguimos sempre pelo menos algum caso em que a acurácia foi 100%, ou seja, não tivemos nenhum erro na fase de teste. Além disso, para todos os classificadores, sempre encontramos, fixado $\sigma = 10$, um parâmetro C e uma combinação de conjunto de treinamento que não apresente nenhum erro. Para estes casos, os conjuntos que sempre apareceram foram $TR2$ e $TR8$, já os parâmetros foram $C = 10^i, i = 2, 3$.

Para obtermos a acurácia, e conseqüentemente o erro, usamos a função *svmtrain* do Matlab. Isto porque seu processo de classificação é mais rápido em comparação ao *quadprog* e a interface do *Algencan*. Posteriormente, usamos a função *svmclassify*, também do Matlab, que fornece as saídas previstas pelo classificador, em nosso conjunto de teste. No Apêndice A, descrevemos um pouco sobre tais funções.

Agora, utilizamos os melhores classificadores, obtidos com $\sigma = 10$, $TR8$ e $C = 10^2$, para classificar a placa de um carro, apresentada na Figura 3.14(d), pela estratégia *one-against-one*. O reconhecimento de uma placa veicular é um pequeno desafio nessa área, devido aos diferentes tipos, formas e cores e também dos diversos meios de exposição luminosa, ângulo e distância durante aquisição da imagem, além de variações do clima, reflexos e movimentos.



(d) Placa inicial, para classificação

(e) Placa do tipo oficial, para classificação.

Entretanto, como o fundo da placa é cinza, não conseguimos classificar corretamente as letras. Em [30] utiliza-se o OpenCV e Tesseract OCR, que localiza as letras e transforma o fundo em branco. Mas, resolvemos já usar uma placa com texto preto e fundo branco, que é usada em carros oficiais (governos, polícias, Corpo de Bombeiros), apresentada na Figura 3.14(e).

Assim, recortamos manualmente cada letra e colocamos em uma cédula do Excel com as mesmas propriedades que usamos para criar as letras do banco de dados. Depois, salvamos essa cédula como imagem e classificamos, obtendo classificação correta apenas para a letra A.

Deste modo, resolvemos testar a estratégia *one-against-rest*, apenas para $TR8$, $\sigma = 10$ e $C = 10^2$, obtendo classificação correta para todas as letras.

Capítulo 4

Conclusões e trabalhos futuros

Neste capítulo, apresentamos as conclusões desta dissertação, bem como as sugestões para trabalhos que futuramente podem ser desenvolvidos.

O foco deste trabalho foi realizar uma análise teórica sobre uma técnica de Aprendizagem de Máquina muito utilizada: as Máquinas de Vetores Suporte (SVM, do Inglês *Support Vector Machine*). Tal técnica, fundamentada em teorias fortes e não-triviais de Otimização e Estatística [21, 22, 34], é muito utilizada em problemas de classificação, por isso fizemos uma pequena aplicação prática em classificação de caracteres.

Provamos conceitos e teorias essenciais da técnica, procurando apresentar da forma mais didática possível. Fundamentamos a teoria da existência e unicidade para obtenção do hiperplano ótimo. Apresentamos uma análise de duas definições de vetores suporte que não são equivalentes. Provamos que ao retirar pontos que não são vetores suporte o classificador não muda. Isto propiciou a proposta de um algoritmo que gera uma quantidade consideravelmente pequena de vetores suporte, além da possibilidade de trabalhar com a interface do Algencan em Matlab [29, 31].

No Capítulo 1 apresentamos conceitos de Análise e Álgebra Linear, em particular sobre espaços de Hilbert. Abordamos otimização convexa, programação quadrática e dualidade pois queríamos justificar a existência de minimizadores. Apresentamos exemplos em que só a convexidade não garante esta existência, nem mesmo com a hipótese de limitação da função objetivo. Tratamos algumas hipóteses, apresentadas no Teorema 1.31, para garantir a unicidade de um problema convexo. Trabalhamos no texto de Blum e Oettli [23], tornando acessível um resultado que não é tão clássico. Para tratarmos da dualidade, reformulamos alguns resultados, para não usarmos ponto de sela.

No Capítulo 2 apresentamos uma revisão de conceitos que embasa tal técnica. Tratamos da forma mais simples de SVM, que é a margem rígida, utilizada na geração de um classificador para conjunto linearmente separável. Depois, estendemos os conceitos abordados para conjuntos mais gerais, através de uma relaxação das restrições impostas na margem de separação entre os dados. Para este problema, reestruturamos os enunciados

e demonstrações em [25]. Tratamos sobre o cálculo de b quando não temos unicidade da solução primal, que sentimos falta em [3, 15, 20]. Porém, o uso da regularização não é suficiente para tratar de todos os problemas envolvendo classificação. Deste modo, tratamos das funções *Kernel*.

Apresentamos duas definições de funções *Kernel*, uma delas usando mapeamento para um espaço de dimensão finita, a outra, mais geral, considerando mapeamentos em espaços de Hilbert. Mostramos que estas duas definições não são equivalentes. Discutimos diversas propriedades, bem como formas de caracterizar uma função *Kernel*, em particular o Teorema de Mercer.

Esta dissertação, além de apresentar um estudo teórico, também abordou o comportamento de SVM em um problema prático, que é a classificação de caracteres. No Capítulo 3, mostramos os testes deste problema. Tentamos detalhar os principais passos. Os resultados alcançados demonstraram a eficácia desta técnica, pois atingiram entre 80% e 100% de acerto, recorrendo ao *Kernel* Gaussiano, variando os parâmetros σ e C e o conjunto de treinamento.

As principais vantagens de SVM, observadas no trabalho, são: possibilidade de lidar com grandes conjuntos de dados e tratamento de dados de alta dimensão. Já sobre as desvantagens, podemos citar: difícil escolha da função *Kernel* e seus parâmetros e principalmente do parâmetro de regularização. Além disso, o tempo de treinamento é longo, dependendo do número e dimensão dos dados.

O trabalho realizado nos trouxe ainda outros pontos que podem ser abordados em trabalhos futuros. A seguir são descritas algumas sugestões:

- Fazer um estudo, conforme elaborado nesta dissertação, buscando verificar a existência e unicidade das soluções, ou condições para tal, das extensões de SVM. Entre elas, v -SVM, que utiliza um parâmetro v para possibilitar um controle efetivo no número de vetores suporte, ou *Leave-one-out* SVM, que é baseada na generalização de margens que podem adaptar-se de forma iterativa.
- Analisar se o Algoritmo 2.1 continua válido para SVM com margem flexível e com a utilização do Truque do *Kernel*. Caso não, quais as possíveis alterações dele. Além disso, encontrar um bom critério de parada.
- Realizar testes, com o mesmo banco de dados usado na dissertação, para verificar a relação de vetores suporte e a acurácia. Além disso, analisar a influência na quantidade de pontos do conjunto de treinamento, para a acurácia.
- Desenvolvimento de algoritmos que melhorem o custo computacional ou que utilizem algumas propriedades dos problemas. Uma possibilidade é estudar o *Sequential Minimal Optimization*, que é o *default* nos programas de SVM do Matlab, utilizado no trabalho.

Apêndice A

Códigos em Matlab

Considerando que não foi objetivo deste trabalho construir um programa competitivo para Máquinas de Vetores Suporte e sim realizar um estudo teórico e avaliar experimentalmente a aplicabilidade ao problema de classificação de caracteres, apresentamos uma possibilidade de implementação em Matlab, para tal técnica.

Primeiramente, descrevemos os principais comandos utilizados para resolução dos problemas propostos no texto, através do quadprog do Matlab. Posteriormente, apresentamos as alterações realizadas nos arquivos *.m situados em /algencan-2.4.0/sources/interfaces/matlab, para utilização da interface do Algencan em Matlab. Por fim, tratamos sobre as funções usadas na Seção 3.2.

Utilização do quadprog

O programa quadprog do Matlab, resolve o problema

$$\begin{aligned} \min_z \quad & \frac{1}{2} z^T H z + f^T z \\ \text{s.a} \quad & A z \leq c, \\ & Aeq z = ceq, \\ & LB \leq z \leq UB, \end{aligned}$$

com $z, f, LB, UB \in \mathbb{R}^n$, $c \in \mathbb{R}^m$, $Aeq \in \mathbb{R}^{1 \times m}$, $ceq \in \mathbb{R}$ e as matrizes H e A possuem dimensões apropriadas de acordo com a dimensão de z . Para obter a solução, basta usar o comando $[z, fval, exitflag] = quadprog(H, f, A, c, Aeq, ceq, LB, UB)$. As saídas $fval$ e $exitflag$ retornam o valor da função no ponto mínimo e as condições do problema, obtidas pelo programa, respectivamente.

A seguir, apresentamos como podemos usar o quadprog para resolver os problemas da SVM, tratados no Capítulo 2.

Resolução do problema primal com margem rígida

Para resolvermos o problema (2.3), podemos reescrevê-lo como

$$\begin{aligned} \min_z \quad & \frac{1}{2}z^T Hz + f^T z \\ \text{s.a} \quad & Az \leq c, \end{aligned}$$

$$\begin{aligned} \text{com } z = \begin{bmatrix} w \\ b \end{bmatrix} \in \mathbb{R}^{n+1}, f = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{n+1}, A = \begin{bmatrix} -y_1(x^1)^T & -y_1 \\ \vdots & \vdots \\ -y_m(x^m)^T & -y_m \end{bmatrix} \in \mathbb{R}^{m \times (n+1)}, \\ c = \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix} \in \mathbb{R}^m \text{ e } H = \begin{bmatrix} I_{n \times n} & 0_{n \times 1} \\ 0_{1 \times n} & 0 \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}. \end{aligned}$$

Para obter o resultado, usamos o comando $[z, fval, exitflag] = quadprog(H, f, A, c)$, já que o problema acima não possui restrições de igualdade e a variável z não é limitada.

Resolução do problema dual com margem rígida

No Capítulo 2, escrevemos o problema dual do Lagrangiano associado ao problema (2.3), para resolvê-lo, podemos reescrevê-lo como,

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x^i)^T x^j - \sum_{i=1}^m \alpha_i \\ \text{s.a} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, \dots, m. \end{aligned} \tag{A.1}$$

Para facilitar na implementação, reescrevemos o problema (A.1) do seguinte modo,

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T H \alpha - e^T \alpha \\ \text{s.a} \quad & y^T \alpha = 0, \\ & \alpha \geq 0, \end{aligned}$$

com $H_{ij} = y_i y_j (x^i)^T x^j$ e e o vetor com entradas iguais a 1. Assim, tomamos $f = -e$, $Aeq = y^T$, $ceq = 0$, $LB = 0$ e $UB = inf$ e utilizamos o comando $[z, fval, exitflag] = quadprog(H, f, [], [], Aeq, ceq, LB, UB)$. Usamos $[\]$, pois o problema não tem restrições de desigualdade.

Resolução do problema primal com margem flexível

Para resolvermos o problema (2.17), definimos $k = n + 1 + m$ e tomamos a variável de otimização como

$$z = \begin{bmatrix} w \\ b \\ \xi \end{bmatrix} \in \mathbb{R}^k, \text{ com } LB = \begin{bmatrix} -inf \\ \vdots \\ -inf \\ 0_{m \times 1} \end{bmatrix} \in \mathbb{R}^k, UB = \begin{bmatrix} inf \\ \vdots \\ inf \end{bmatrix} \in \mathbb{R}^k,$$

$$H = \begin{bmatrix} I_{n \times n} & 0_{n \times (m+1)} \\ 0_{(m+1) \times n} & 0_{(m+1) \times (m+1)} \end{bmatrix} \in \mathbb{R}^{k \times k}, f = \begin{bmatrix} 0_{(n+1) \times 1} \\ C.e \end{bmatrix} \in \mathbb{R}^k,$$

$$A = \begin{bmatrix} -diag(y)X & -y & -I_{m \times m} \end{bmatrix} \in \mathbb{R}^{m \times k}, c = \begin{bmatrix} -e \end{bmatrix} \in \mathbb{R}^m,$$

onde e é um vetor com todas as componentes iguais a 1. Com X e y dados por

$$X = \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \vdots \\ (x^m)^T \end{bmatrix} \in \mathbb{R}^{m \times n} \text{ e } y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m.$$

Resolução do problema dual com margem flexível

Reescrevemos o problema (2.44) na forma matricial

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T H \alpha - e^T \alpha \\ \text{s.a.} \quad & y^T \alpha = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m, \end{aligned} \tag{A.2}$$

com $H_{ij} = y_i y_j (x^i)^T x^j$ e e o vetor com entradas iguais a 1. Os comandos que usamos para resolver o problema são iguais aos do problema (A.1), exceto que $UB = C.e$.

Resolução do problema primal e dual, com regularização e *Kernel*

Para resolvermos (2.54) tendo o conhecimento da função ϕ , basta trocar x^i por $\phi(x^i)$ nos comandos utilizados para os problemas (2.17). Do mesmo modo, podemos resolver (2.55) trocando $(x^i)^T x^j$ pela função *Kernel* escolhida $K(x^i, x^j)$, nos comandos utilizados para resolver o problema dual com regularização.

Utilização da interface do Algencan

Algencan foi desenvolvido para resolver o problema de programação não linear formulado como

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.a} \quad & h(x) = 0 \\ & g(x) \leq 0 \\ & LB \leq x \leq UB, \end{aligned}$$

com $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ que se supõem deriváveis.

Este programa faz parte do projeto TANGO e pode ser livremente acessado em

<http://www.ime.usp.br/~egbirgin/tango/>.

Pode ocorrer que, ao acessar o programa, não se obtenha exatamente o descrito a seguir, já que a implementação de Algencan tem recebido alterações. Mas enfatizaremos as principais características no decorrer do texto. Para maiores detalhes sobre a instalação e utilização do Algencan, podemos consultar [29, 31], já para conceitos de programação não linear [26].

Para a instalação do Algencan usando a interface em Matlab, o usuário deve baixar o Algencan compactado, da página acima, clicar em ‘outras versões’ e baixar o arquivo ‘algencan-2.4.0.tgz’. Depois, copiar o Algencan descompactado em sua pasta pessoal. Entrar no terminal e digitar ‘make algencan-matlab’ ou ‘make algencan-matlab 64BITS=true’ para plataforma 64 bits. Isso gera um arquivo executável em Matlab, chamado algencan.mexglx em /algencan-2.4.0/bin/matlab. Criar uma pasta fora da pasta algencan-2.4.0 e copiar nela todos os arquivos *.m em /algencan/sources/interfaces/matlab. Isto porque, toda vez que o usuário fechar o Terminal e depois desejar compilar o arquivo para executá-lo, deverá executar novamente este processo, perdendo as alterações feitas em algencan-2.4.0. Entrar no Matlab e substituir \$(ALGENCAN) com o diretório, digitando ‘addpath(‘\$(ALGENCAN)/bin/matlab’)’. Para executar o Algencan digitar ‘algencanma’. Poderão ocorrer alguns erros durante estes passos, pois depende da versão utilizada do Matlab e do sistema operacional (Windows, Linux, etc), em nosso trabalho ocorreram alguns erros nos passos, que só puderam ser solucionados pelo Professor Francisco Nogueira Calmon Sobral da Universidade Estadual de Maringá.

Para realizar as otimizações da SVM, descrevemos os principais comandos de programação contidos em algencan-2.4.0/sources/interfaces/matlab a fim de alterar os arquivos em Matlab da pasta pessoal, para resolver o problema de interesse.

Em algencan-2.4.0/sources/interfaces/matlab/inip.m, aparece $n = 2$, isto significa que temos duas variáveis (x_1 e x_2). Mais abaixo, temos $x = \text{zeros}(n, 1)$, ou seja, o valor inicial das variáveis é zero. Temos também a limitação das variáveis

```
for i = 1 : n - 1
    l(i) = -10;
```

```

    u(i) = 10;
end
l(n) = -1.0d + 20;
u(n) = 1.0d + 20;

```

A notação $-1.0d + 20$ representa -1×10^{20} e as letras l e u significam *lower* e *upper*, respectivamente. O equivalente matematicamente é

$$-10 \leq x_1 \leq 10 \text{ e } -10^{20} \leq x_2 \leq 10^{20}.$$

A limitação de x_2 entre -10^{20} e 10^{20} é para deixar esta variável livre, pois a teoria do Algencan exige que as variáveis estejam em um compacto. O número de restrições neste caso é dado por $m = 2$. A linha de comando `equatn = zeros(m, 1)` significa que as restrições de 1 até m não são de igualdade. Em Algencan, as desigualdades são sempre consideradas da forma menor ou igual a zero.

Como para cada restrição é vinculado um valor de *lambda*, multiplicador de Lagrange, o comando `lambda = zeros(m, 1)` significa que os m multiplicadores iniciam iguais a zero. As linhas `linear(1) = 0` e `linear(2) = 1` informam a linearidade das restrições, em que a primeira restrição é não linear e a segunda é. No que segue, o usuário deve informar quais sub-rotinas estarão sendo passadas para o programa, deixando as ativas, ou seja, que serão usadas no programa, com o comando `coded(.) = 1`. A linha `checkder = 1` pede ao programa que o valor das derivadas informadas seja conferido através do método das diferenças finitas.

A primeira subrotina passada ao Algencan chama-se *evalf* e é nela onde o usuário informa qual é a função objetivo do problema de otimização, descrita na linha `f(x) = x(2)`. Suas restrições são passadas na subrotina *evalco*, dadas nas linhas `c(1) = x(1)^2 + 1 - x(2)` e `c(2) = 2 - x(1) - x(2)`. Na subrotina *evalg*, o usuário deve informar o gradiente relativo à função objetivo, no exemplo, `g(1) = 0` e `g(2) = 1`, ou seja, a derivada da função objetivo em relação à primeira variável é zero e em relação à segunda variável é 1.

Matematicamente, o problema em `algencan-2.4.0/sources/interfaces/matlab` é

$$\begin{array}{ll}
 \min_x & x_2 \\
 \text{s.a} & x_1^2 + 1 - x_2 \leq 0, \\
 & 2 - x_1 - x_2 \leq 0, \\
 & -10 \leq x_1 \leq 10, \\
 & -10^{20} \leq x_2 \leq 10^{20}.
 \end{array}$$

A partir daqui podemos modificar os arquivos, a fim de resolver nosso próprio problema de otimização.

No que segue, estamos considerando *ytr* como as saídas do conjunto de treinamento e *TR* uma matriz cujas linhas são os dados de treinamento transpostos. Então,

precisamos primeiramente declará-las globais no arquivo algencanma.m.

Resolução do problema primal com margem rígida

Para a resolução do problema (2.3), modificamos no arquivo inip.m a limitação, a linearidade das restrições, o número de variáveis e de restrições, conforme a quantidade e dimensão dos dados de treinamento, deixamos ativas apenas as subrotinas evalf.m, evalg.m e evalco.m.

Tomamos $n = n + 1$, ou seja, a dimensão da solução x é a dimensão de $w \in \mathbb{R}^n$ mais a dimensão de $b \in \mathbb{R}$. Assim, $x(1 : n - 1) = w$ e $x(n) = b$.

% Limitantes - Note que as variáveis são livres.

for i = 1 : m

l(i) = -1.0d + 20;

u(i) = 1.0d + 20;

end

% Todas as restrições são lineares.

for i = 1 : m

linear(i) = 1;

end

No arquivo evalco.m alteramos as restrições,

% Restrições de desigualdade.

for i = 1 : m

if (ind == i)

*c = 1 - ytr(i) (TR(i, :) * x(1 : n - 1) + x(n));*

end

end

No arquivo evalg.m alteramos a derivada da função objetivo,

% Derivada da função objetivo em relação a w.

for i = 1 : n - 1

g(i) = x(i);

end

% Derivada da função objetivo em relação a b.

g(n) = 0;

Alteramos também a função objetivo no arquivo evalf.m,

% Função objetivo.

$$f(x) = 1/2 * (x(1 : n - 1)' * x(1 : n - 1));$$

Resolução do problema dual com margem rígida

Para resolvermos o problema A.1, consideramos $x = \alpha$ e modificamos as seguintes linhas em `inip.m`,

`% Número de variáveis` passa a ser a quantidade de dados de treinamento.

```
n = size(TR, 1);
```

`% Número de restrições - Restrição de igualdade e desigualdade.`

```
m = 1 + size(TR, 1);
```

`% Limitantes - Restrições de caixa.`

```
for i = 1 : m
```

```
    l(i) = 0;
```

```
    u(i) = 1.0d + 20; % Sem limitante superior.
```

```
end
```

`% Restrições`

```
    equatn(1) = 1; % Restrição de igualdade.
```

```
for i = 2 : m
```

```
    equatn(i) = 0; % Restrições de desigualdade.
```

```
end
```

`% Restrições Lineares`

```
    linear = ones(m, 1);
```

No arquivo `evalco.m` alteramos as restrições,

`% Restrição`

```
if (ind == 1)
```

```
    c = ytr' * x; % Restrição de igualdade
```

```
end
```

```
for i = 1 : n
```

```
    if (ind == i + 1)
```

```
        c = -x(i); % Restrição de igualdade
```

```
    end
```

```
end
```

No arquivo `evalg.m` alteramos a derivada da função objetivo,

`% Derivada da função objetivo em relação a α`

```
g = H * x - ones(n, 1);
```

Por fim, alteramos a função objetivo no arquivo `evalf.m`,

`% Função objetivo.`

$$f(x) = 1/2 * x' * H * x - ones(1, n) * x,$$

com $H(i, j) = ytr(i) * ytr(j) * TR(i, :) * TR(j, :)'$.

Resolução do problema primal com margem flexível

Agora, para o problema (2.17), consideramos $x = (w, b, \xi)^T$ e modificamos as seguintes linhas em inip.m,

% Número de variáveis é igual a dimensão de w , b e ξ .

```
n = size(TR, 2) + 1 + size(TR, 1);
```

% Número de restrições é igual a quantidade de dados.

```
m = 2 * size(TR, 1);
```

% Limitantes - Variáveis livres w e b .

```
for i = 1 : size(TR, 2) + 1
```

```
    l(i) = -1.0d + 20;
```

```
    u(i) = 1.0d + 20;
```

```
end
```

% Limitação para ξ .

```
for i = size(TR, 2) + 2 : n
```

```
    l(i) = 0;
```

```
    u(i) = 1.0d + 20; % Sem limitante superior.
```

```
end
```

% Restrições de desigualdade.

```
equatn = zeros(m, 1);
```

% Restrições Lineares

```
linear = ones(m, 1);
```

No arquivo evalco.m alteramos as restrições,

% Restrições de desigualdade para w e b

```
for i = 1 : size(TR, 1)
```

```
    if (ind == i)
```

```
        c = 1 - ytr(i) * (TR(i, :) * x(1 : size(TR, 2)) +  
            x(size(TR, 2) + 1)) - x(size(TR, 2) + 2);
```

```
    end
```

```
end
```

% Restrições de desigualdade para ξ

```
for i = size(TR, 1) + 1 : 2 * size(TR, 1)
```

```
    if (ind == i)
```

```
        c = -x(size(TR, 2) + 1 + i - size(TR, 1));
```

```
    end
```

```
end
```

No arquivo evalg.m, alteramos a derivada da função objetivo,

% Derivada em relação a w

```
for i = 1 : size(TR, 2)
```

```
    g(i) = x(i);
```

end

% Derivada em relação a b

$$g(\text{size}(TR, 2) + 1) = 0;$$

% Derivada em relação a ξ

for $i = \text{size}(TR, 2) + 2 : n$

$$g(i) = C;$$

end

A função objetivo é alterada, no arquivo evalf.m, para

$$f(x) = 1/2 * (x(1 : \text{size}(TR, 2))' * x(1 : \text{size}(TR, 2))) + \\ C * \text{ones}(1, \text{size}(TR, 1)) * x(\text{size}(TR, 2) + 2 : n);$$

Resolução do problema dual com margem flexível

Os comandos são iguais aos do problema (A.1), exceto que em inip.m temos

% Limitantes - Restrições de caixa.

for $i = 1 : m$

$$l(i) = 0;$$

$$u(i) = C; \text{ \% } C \text{ Parâmetro de regularização.}$$

end

Resolução do problema primal e dual, com regularização e *Kernel*

Para resolver (2.54) com o Algencan, tendo o conhecimento da função ϕ , basta trocar x^i por $\phi(x^i)$ nos comandos utilizados para o problema (2.17). Além disso, devemos trocar a dimensão de w para N , já que os dados de treinamento serão mapeados pela função ϕ para \mathbb{R}^N .

Assim, para resolvermos (2.54) trocamos em inip.m,

% Número de variáveis

$$n = \text{size}(\text{phi}(TR(1, :)), 1) + 1 + \text{size}(TR, 1);$$

% Limitantes para w e b

for $i = 1 : \text{size}(\text{phi}(TR(1, :)), 1) + 1$

$$l(i) = -1.0d + 20;$$

$$u(i) = 1.0d + 20;$$

end

% Limitantes para ξ

for $i = \text{size}(\text{phi}(TR(1, :)), 1) + 2 : n$

$$l(i) = 0;$$

$$u(i) = 1.0d + 20;$$

end

Em evalg.m e evalf.m trocamos

% Derivada da função objetivo em relação a w

```

for i = 1 : size(phi(TR(1, :)), 1)
    g(i) = x(i);
end
% Derivada da função objetivo em relação a b
g(size(phi(TR(1, :)), 1) + 1) = 0;
% Derivada da função objetivo em relação a ξ
for i = size(phi(TR(1, :)), 1) + 2 : n
    g(i) = C;
end

A função objetivo é alterada, no arquivo evalf.m, para

$$f(x) = 1/2 * (x(1 : size(phi(TR(1, :)), 1))' * x(1 : size(phi(TR(1, :)), 1)) + C * ones(1, size(TR, 1)) * x((phi(TR(1, :)), 1) + 2 : n));$$

Por fim, em evalco.m tomamos
% Restrições de desigualdade para w e b.
for i = 1 : size(TR, 1)
    if (ind == i)
        c = 1 - ytr(i) * (phi(TR(i, :))' * x(1 : size(phi(TR(1, :)), 1)) + x(size(phi(TR(1, :)), 1) + 1)) - x(size(phi(TR(1, :)), 1) + 2);
    end
end
% Restrições de desigualdade para ξ.
for i = size(TR, 1) + 1 : 2 * size(TR, 1)
    if (ind == i)
        c = -x(size(phi(TR(1, :)), 1) + 1 + i - size(TR, 1))
    end
end

```

end

Agora para resolvermos (2.55), usaremos os mesmos comandos do problema dual com margem flexível, exceto que trocamos o produto $(x^i)^T x^j$ pela função *Kernel* escolhida $K(x^i, x^j)$.

Comandos Linux

Em nosso trabalho, utilizamos a Interface em Matlab do Algencan-2.4.0, no Linux. Assim, listamos abaixo alguns dos seus comandos e respectivos significados.

- cd - utilizado para a navegação entre pastas do computador. Acessa e muda de diretório corrente. Se acrescentar dois pontos finais, acessamos o nível superior.
- ls - utilizado para listar os arquivos contidos dentro do diretório corrente.
- cp - utilizado para copiar arquivos.

- pwd - utilizado para imprimir o nome do diretório onde o usuário se encontra.
- make - utilizado para ler e compilar o arquivo “Makefile” e compilar arquivos.
- mkdir - utilizado para criar um diretório dentro do qual o usuário está.

Para obtermos os resultados no problema de classificação de caracteres, descrito no Capítulo 3, usamos algumas funções do Matlab, em sua versão R2015a, próprias para a SVM. Assim, no que segue descrevemos resumidamente tais funções.

svmtrain e svmclassify

Para utilizar a `svmtrain` digitamos o seguinte comando, que retorna uma estrutura, `SVMStruct`, contendo informações sobre o classificador de SVM.

```
» SVMStruct = svmtrain(Treinamento,Grupo).
```

Argumentos de entrada:

- `Treinamento`: Matriz com dados de treinamento, em que cada linha corresponde a uma observação e cada coluna corresponde a uma característica.
- `Grupo`: Variável de agrupamento, que pode ser um vetor categórico, numérico ou lógico, uma matriz de caracteres de vetores de caracteres ou uma matriz de caracteres com cada linha que representa um rótulo de classe. Cada elemento do `Grupo` especifica o grupo da linha correspondente de treinamento. O grupo deve dividir o treinamento em dois grupos.

Para obtermos as saídas previstas, pelo classificador obtido com `svmtrain`, usamos o comando

```
» Group = svmclassify(SVMStruct,Sample).
```

Classifica cada linha dos dados na amostra, uma matriz de dados, usando as informações em uma estrutura de classificador de máquinas vetoriais de suporte `SVMStruct`, criadas usando a função `svmtrain`. Como os dados de treinamento usados para criar `SVMStruct`, `Sample` é uma matriz em que cada linha corresponde a uma observação e cada coluna corresponde a uma característica ou variável. O `Grupo` indica o grupo ao qual cada linha de amostra foi atribuída.

Argumentos de entrada:

- `SVMStruct`: Estrutura do classificador de máquina vetorial criada usando a função `svmtrain`.
- `Sample`: Uma matriz em que cada linha corresponde a uma observação ou a uma repetição, e cada coluna corresponde a uma característica ou variável.

Referências Bibliográficas

- [1] V. T. Ales. *O Algoritmo Sequential Minimal Optimisation para Resolução do Problema de Support Vector Machine: uma Técnica para Reconhecimento de Padrões*. Dissertação de Mestrado em Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, 2008.
- [2] M. Beltrami. *Método Grid-Quadtree para seleção de parâmetros do algoritmo support vector classification (SVC)*. Tese de Doutorado em Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, 2016.
- [3] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [4] C. J. C. Burges. *A tutorial on support vector machines for pattern recognition*. Data mining and knowledge discovery. Springer. v. 2, n. 2, pp. 121-167, 1998.
- [5] B. Carvalho. *Novas estratégias para detecção automática de vetores de suporte em least squares support vector machines*. Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal de Minas Gerais, 2005.
- [6] D. S. G. Carvalho. *Sobre Máquinas de Vetores Suporte para classificação de Caracteres*. Monografia (Bacharel em Matemática Industrial), Universidade Federal do Paraná, Curitiba, 2013.
- [7] T. S. de Almeida et al. *Algoritmo para detecção de boca em faces humanas usando matriz de Co-ocorrência e SVM*. Cadernos de Pesquisa, v. 19, 2012.
- [8] A. C. dos Santos. *Máquinas de suporte vetorial e sua aplicação na detecção de spam*. Trabalho de Formatura Supervisionado, Universidade de São Paulo, Instituto de Matemática e Estatística, 2005.
- [9] A. C. Lorena e A. de Carvalho. *Uma introdução às support vector machines*. Revista de Informática Teórica e Aplicada. v. 14. n. 2, pp. 43-67, 2007.
- [10] B. Schölkopf e A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [11] M. R. Pinheiro e C. Grande. *Conjugação e dualidade em programação convexa*. Relatório técnico, 1984.

- [12] C. Chang e C. Lin. {LIBSVM}: *a Library for Support Vector Machines (Version 2.3)*, 2001.
- [13] H. Andrews e C. Patterson. *Singular value decomposition (SVD) image coding*. IEEE transactions on Communications. v. 24. n. 4, pp. 425-432, 1976.
- [14] A. A. Ribeiro e E. W. Karas. *Otimização Contínua: Aspectos teóricos e computacionais*. Cengage Learning, Brasil, 2013.
- [15] N. Cristianini e J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.
- [16] N. Cristianini e J. Shawe-Taylor. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.
- [17] A. Izmailov e M. Solodov. *Otimização: Condições de Otimalidade, Elementos de Análise Convexa e Dualidade*. v. 1. IMPA, Rio de Janeiro, 2005.
- [18] M. Frank e P. Wolfe. *An Algorithm for Quadratic Programming*. Naval Research Logistics Quarterly. v. 3. pp. 95-110, 1956.
- [19] J. Nocedal e S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 2nd edition, 2006.
- [20] I. Griva e S. Nash. *Linear and nonlinear optimization*. SIAM, 2009.
- [21] C. Cortes e V. N. Vapnik. *Support-vector networks*. Machine learning. Springer. v. 20. n. 3, pp. 273-297, 1995.
- [22] I. M. Guyon e V. N. Vapnik. *A training algorithm for optimal margin classifiers*. Proceedings of the fifth annual workshop on Computational learning theory. ACM. pp. 144-152, 1992.
- [23] E. Blum e W. Oettli. *Technical Note – Direct proof of the Existence Theorem for Quadratic Programming*. Operations Research. v. 20. n. 1, pp. 165-167, 1972.
- [24] A. C. Lorena et al. *Inteligência Artificial: Uma abordagem de aprendizado de máquina*. Rio de Janeiro: LTC, v. 2. pp. 192, 2011.
- [25] C. J. C. Burges et al. *Uniqueness of the SVM Solution*. NIPS. v. 99, pp. 223-229, 1999.
- [26] A. Friedlander. *Elementos de Programação Não-Linear*. Unicamp, 1994.
- [27] Chih-Wei et al Hsu. *A practical guide to support vector classification*. Taipei, Taiwan, 2003.

- [28] A. R. G. Lima. *Máquinas de vetores suporte na classificação de impressões digitais*. Universidade Federal do Ceará, Fortaleza, 2002.
- [29] J. M. Martínez. *Otimização prática usando o Lagrangiano Aumentado*. Departamento de Matemática Aplicada, Universidade Estadual de Campinas, Brasil, 2006.
- [30] J. D. do Nascimento. *Detecção e reconhecimento de placa automotiva com baixo custo*. Monografia. Centro Universitário de Brasília, 2012.
- [31] R. Penachi et al. *Uma introdução à otimização não linear e a solução de problemas simétricos via ALGENCAN*. Campinas - SP, 2015.
- [32] V. L. Rodrigues. *Reconhecimento facial usando SVM*. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática, 2007.
- [33] R. A. Scarpel. *Utilização de Support Vector Machine em previsão de insolvência de empresas*. Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional. pp. 671-677, 2005.
- [34] V. N. Vapnik. *Pattern recognition using generalized portrait method*. Automation and remote control. v. 24. pp. 774-780, 1963.
- [35] A. Verdério. *Sobre o uso de regressão por vetores suporte para a construção de modelos em um método de região de confiança sem derivadas*. Tese de Doutorado. Universidade Federal do Paraná, Curitiba, 2015.