UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

CAIO VINICIUS DADAUTO

Método das Direções Alternadas para Multiplicadores com Aplicações em Problemas de Suporte Vetorial Distribuído

Campinas

2018

Caio Vinicius Dadauto

Método das Direções Alternadas para Multiplicadores com Aplicações em Problemas de Suporte Vetorial Distribuído

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Matemática Aplicada.

Orientador: Paulo José da Silva e Silva

Este exemplar corresponde à versão final da Dissertação defendida pelo aluno Caio Vinicius Dadauto e orientada pelo Prof. Dr. Paulo José da Silva e Silva.

Campinas

2018

Agência(s) de fomento e nº(s) de processo(s): CNPq, 132125/2016-1
ORCID: https://orcid.org/0000-0002-0986-007X

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

D12m

Dadauto, Caio Vinicius, 1990-
Método das direções alternadas para multiplicadores com aplicações em problemas de suporte vetorial distribuído / Caio Vinicius Dadauto. – Campinas, SP : [s.n.], 2018.

Orientador: Paulo José da Silva e Silva.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Otimização matemática. 2. Programação convexa. 3. Aprendizado de máquina. I. Silva, Paulo José da Silva e, 1973-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Alternating directions method of multipliers with applications in distributed support vector problems
Palavras-chave em inglês:
Mathematical optimization
Convex programming
Machine learning
Área de concentração: Matemática Aplicada
Titulação: Mestre em Matemática Aplicada
Banca examinadora:
Paulo José da Silva e Silva [Orientador]
Jose Mario Martinez Perez
Maicon Marques Alves
Data de defesa: 19-03-2018
Programa de Pós-Graduação: Matemática Aplicada

Dissertação de Mestrado defendida em 19 de março de 2018 e aprovada pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). PAULO JOSÉ DA SILVA E SILVA

Prof(a). Dr(a). JOSE MARIO MARTINEZ PEREZ

Prof(a). Dr(a). MAICON MARQUES ALVES

As respectivas assinaturas dos membros encontram-se na Ata de defesa

Para minha família, meu amor e meus amigos ...

Agradecimentos

Inicialmente agradeço aos meus pais, *Janete* e *Arthur*, por me apoiarem mesmo quando optava por caminhos incertos, por confiarem no meu senso de julgamento e, principalmente, pelo amor incondicional que sempre demonstraram por mim. Se hoje consigo vencer meus próprios obstáculos, devo isso aos meus pais que são os meus maiores exemplos de superação. Ainda, agradeço ao meu amor, *Bruna Lopes*, que com gestos carinhosos e conversas infindáveis me trouxe paz e lucidez quando o cansaço e o desespero me dominavam.

Ademais, sou grato aos meus amigos que fizeram ou ainda fazem parte da minha formação e, por isso, participaram diretamente ou não desta dissertação, são eles *Feio, San, Bill, Mark, Laura, Deivid, Daniel, Filipe, Bin, Vinão, Mirtão, Nagaoka, Russo, Rogerião, Alex, Lima* ...

Além disso, agradeço ao meu orientador, *Paulo*, por confiar em mim e por estar sempre disposto a realizar sugestões e correções que foram decisivas para a finalização desta dissertação. Por fim, agradeço a *CNPq* por financiar este projeto de Mestrado.

"Do conhecimento provo, não me privo
Me torno mais livre a cada livro, me livro
Do pensamento mais fútil, da cultura inútil
Que não passa pelo crivo"
(Fabio Brazza - Sem Moda, Sem Medo)

Resumo

Essa dissertação trata do paradigma de *máquina de suporte vetorial (SVM)* aplicado a um problema de aprendizado supervisionado, com o conjunto de treino distribuído entre agentes de uma rede conexa em que a comunicação dos dados de treino é proibida. Para tanto, o *SVM* é reescrito como um conjunto de subproblemas de otimização convexa que podem ser computados por cada agente de forma síncrona, bastando para isso apenas a comunicação de determinados vetores, os quais são de consenso a toda rede. Esses subproblemas são introduzidos através da aplicação do *método de direções alternadas para multiplicadores (ADMM)*. Visto a importância desse método e desse paradigma à abordagem distribuída do *SVM*, os aspectos teóricos relevantes ao *ADMM* e ao *SVM* são tratados de forma rigorosa no início dessa dissertação. São apresentadas duas variantes para o *SVM* distribuído, sendo uma para a classificação linear e outra para a classificação não linear. Por fim, são realizadas simulações numéricas que justificam o uso do método proposto. De fato, o método apresenta, para ambas as variantes, capacidade preditiva superior a aplicação do *SVM* apenas sobre os dados de um único agente sem a colaboração com toda a rede. Ademais, a variante para a classificação linear apresenta um modelo tão acurado quanto o modelo proveniente da aplicação do *SVM* linear a todo conjunto de dados.

Palavras-chave: Otimização Matemática; Programação Convexa; Aprendizado de Máquina.

Abstract

This dissertation deals with the *support vector machine (SVM)* classifier applied to a supervised learning problem where the training data is distributed among the agents of a connected network that can not to share all information. To achieve this, the *SVM* is rewritten as a set of convex optimization subproblems that can be computed synchronously for each agent. Those subproblems are introduced by the *alternating direction method of multipliers (ADMM)*. As these ideas are important to distributed *SVM* approach, the main theoretical aspects for *ADMM* and *SVM* are formally discussed in the beginning of this dissertation. Two variants of distributed *SVM* are proposed, namely the approach to linear and another for nonlinear classification. Lastly, numerical simulations that justify the use of the proposed method. In fact, the method shows, for both variants, greater predictive capacity than the application of the *SVM* to the data from only one agent without the collaboration with the whole network. Moreover, the variant for linear classification shows a model that is as accurate as the model from the application of the linear *SVM* in the whole data set.

Keywords: Mathematical Optimization; Convex Programming; Machine Learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ADMM	Alternating Direction Method of Multipliers
SVM	Support Vector Machine
KKT	Condições de Karush-Kuhn-Tucker
LDSVM	Linear Distributed Support Vector Machine
NDSVM	Nonlinear Distributed Support Vector Machine
RHS	Reproducing Hilbert Space
RKHS	Reproducing Kernel Hilbert Space

Lista de símbolos

δ_{ij}	Delta de *Kronecker*
$f^*(\cdot)$	Função conjugada
∇f	Gradiente da função f
$\mathcal{L}$	Lagrangiano
$\mathcal{L}_c$	Lagrangiano aumentado
$\otimes$	Produto de *Kronecker*
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	Produto interno definido no espaço $\mathcal{H}$
$\partial f(\cdot)$	Subdiferencial de f
$\text{sign}(f(x))$	Sinal de f avalidado em x

Lista de Algoritmos

Sumário

Introdução

Em ambientes onde diversos dispositivos estão conectados em rede, a necessidade de extrair informações provenientes de um conjunto de dados é cada vez mais recorrente em vários problemas práticos. Contudo, devido ao crescente volume de dados coletados por um espectro enorme de sensores [Jerzak e Ziekow (2014)], é impraticável a manipulação por intervenção humana desses dados com o intuito de obter as informações de interesse.

Procurando superar a ineficiência intrínseca à análise manual dos dados, é de interesse minimizar a intervenção humana nesse processo de extração de informação. Assim, são empregadas soluções complexas e refinadas provenientes do aprendizado de máquina. Entretanto, apesar dessas soluções minimizarem a intervenção humana, há a exigência de que esses dados estejam centralizados em um único computador, o que, muitas vezes, é impraticável ou até proibido. Neste contexto, há inúmeras aplicações de interesse, a saber

Quanto ao Volume de Dados Aplicações que demandam volumes enormes de dados tornando seu armazenamento em uma única unidade de processamento impraticável. Como exemplo, o acelerador de partículas *LHC* faz uso de técnicas de aprendizado de máquina para identificar eventos de interesse entre os eventos coletados. Contudo, são registrados bilhões de eventos por segundo, o que pode gerar petabytes de dados por mês [Shiers (2007)].

Quanto a Privacidade Aplicações em que os dados não podem ser comunicados entre os agentes da rede por uma questão de privacidade, apesar da necessidade de processar todos os dados distribuídos nessa rede. Como exemplo, há a criação de modelos para um diagnósticos autônomo a partir dos dados provenientes de instituições hospitalares conectadas em rede. Neste caso, os dados de cada instituição não podem ser compartilhados devido a privacidade inerente a esses dados [Scardapane et al. (2018)].

Quanto ao Custo de Recursos Aplicações em que há demasiado custo computacional para comunicação ou em que a capacidade de processamento de cada agente é limitada. Como exemplo, a extração de informação em sensoriamento remoto, caso em que os sensores, normalmente, são de baixo custo monetário, fato que acarreta em um processamento limitado. Ainda, pode-se considerar o sensori-

amento remoto em meios aquosos, neste caso a comunicação exige demasiado processamento e custo energético [Sanford, Potkonjak e Slijepcevic (2012)].

Quanto a Competição e Cooperação Aplicações em que os agentes competem entre si, porém sobre um certo aspecto é vantajoso à todos os agentes que haja uma cooperação mútua. Como exemplo, a criação de modelos para a análise de crédito em instituições financeiras [Louzada, Ara e Fernandes (2016)]. Neste caso, é de interesse à estas instituições a criação de um modelo baseado no banco de dados de todas elas, contudo esses dados não podem ser comunicados devido a possibilidade de favorecimento de uma instituição em detrimento a outra.

Portanto, busca-se determinar algoritmos que solucionam os problemas de aprendizado de máquina de forma distribuída limitando a comunicação ao mínimo necessário para convergir o algoritmo a um modelo preditivo para cada agente da rede, de forma que o modelo leve em consideração os dados de cada agente, porém sem que os dados sejam comunicados entre os mesmos. Em última análise, a maioria desses algoritmos procuram determinar versões distribuídas para o problema

$$\min_{x \in X} \left\{ \sum_{i=0}^{M} f_i(x) \right\}, \tag{1}$$

em que X é um conjunto convexo e f_i é uma função convexa. Essa classe de problemas se enquadra em diversas formulações de modelos pertinentes ao aprendizado de máquina [Boyd et al. (2010)], por exemplo, a máquina de suporte vetorial (*SVM*).

Nesse entendimento, em Forero, Cano e Giannakis (2010) é apresentada uma versão distribuída do *SVM* aplicado a uma rede conexa e descentralizada. Este algoritmo faz uso do método das direções alternadas para multiplicadores (*ADMM*) alinhado a restrições que impõem o consenso entre as soluções derivadas por cada agente da rede. Esta dissertação detalha os aspectos teóricos por trás dessa abordagem distribuída do *SVM* e aborda de forma criteriosa sua derivação. Por fim, são realizadas simulações numéricas para averiguar a eficácia desse método.

I

Contexto Teórico

CAPÍTULO 1

Otimização

A teoria de otimização é de suma importância à aprendizagem de máquina. Em particular, o método ADMM possui grande relevância no contexto distribuído. Tal método pode ser interpretado como uma versão radical do método de lagrangiano aumentado inexato. Portanto, serão discutidos aspectos teóricos intrínsecos à compreensão do mesmo, tais como o problema dual, proximal e dual proximal e, posteriormente, será introduzido o ADMM.

1.1 Notação e definições

Algumas notações serão utilizadas durante o trabalho, a saber, as desigualdades vetoriais dadas por

$$v \begin{array}{c} \leqslant \\ \geqslant \\ < \\ > \end{array} u,$$

com v, $u \in \mathbb{R}^n$, serão interpretadas, para todo $i = 1, \cdots, n$, como

$$v_i \begin{array}{c} \leqslant \\ \geqslant \\ < \\ > \end{array} u_i,$$

com v_i e u_i coordenadas dos vetores v e u, respectivamente.

Os vetores nulos, serão denotados simplesmente por 0, o qual estará inserido no espaço real que convêm. Por exemplo, seja $u \in \mathbb{R}^n$, então a igualdade

$$u = 0$$

denota $0 \in \mathbb{R}^n$. De forma análoga, nas matrizes identidade serão denotadas apenas por I, sem índices que evidenciem o espaço em que estão inseridas.

Com respeito a norma de vetores, a norma 2 será denotada apenas pelo simbolo $||\cdot||$. Por outro lado, a norma $p \geqslant 1$ será denotada por um índice, a saber $||\cdot||_p$. Quaquer outra norma será definida em momento oportuno.

Ademais, serão definidos alguns conceitos pertinentes ao desenvolvimento da dissertação. A saber, as definições de função própria e semi-contínua inferior.

Definição 1. *Uma função $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{-\infty, \infty\}$ é dita própria se*

$$f(x) < \infty$$

ao menos para um $x \in \mathbb{R}^n$, e

$$f(x) > -\infty$$

para todo x de seu domínio.

Definição 2. *Uma função $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{-\infty, \infty\}$ é dita semi-contínua inferior se*

$$f(\bar{x}) \leqslant \liminf_{l \to \infty} f(x_l),$$

em que $\{x_l\}$ é uma sequência qualquer convergente à $\bar{x}$.

Outra definição de interesse é a de subgradiente, como segue abaixo.

Definição 3. *Um subgradiente de uma função $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{-\infty, \infty\}$ convexa não necessariamente diferenciável em x é um vetor $g \in \mathbb{R}^n$ tal que*

$$f(u) \geqslant f(x) + g^T(u - x) \;\; \forall u \in \mathbb{R}^n.$$

Ademais, o conjunto de subgradientes que satisfazem essa desigualdade em x é denominado subdiferencial de f em x e é denotado por

$$\partial f(x).$$

Por fim, define-se a função conjugada.

Definição 4. *Seja a função $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{-\infty, \infty\}$ própria, convexa e semi-contínua inferior. A função conjugada de f é definida por*

$$f^*(y) = \sup_{x \in \mathbb{R}^n} \{y^T x - f(x)\},$$

com $y \in \mathbb{R}^n$.

Durante o desenvolvimento da dissertação, a notação * é utilizada tanto para se referenciar a função conjugada quanto para se referenciar a soluções de um problema de minimização ou de maximização. Apesar do abuso de notação, o contexto em que cada notação se encontra deixa claro ao que esta se refere.

1.2 DUALIDADE LAGRANGIANA

Será definido o problema dual e seus principais resultados, dualidade fraca e forte, a partir de resultados algébricos[1].

A priori, define-se o problema primal de interesse. A saber, sejam $f : \mathbb{R}^n \mapsto \mathbb{R}$, $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ e $h : \mathbb{R}^n \mapsto \mathbb{R}^l$ funções semi-contínuas inferiores não necessariamente convexas, assim, o problema de interesse é dado por

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & h(x) = 0 \\ & g(x) \leqslant 0 \\ & x \in \mathbb{R}^n. \end{aligned} \tag{1.1}$$

Ademais, o conjunto de soluções ótimas de (1.1) será denotado por X^*. Considera-se que X^* é não vazio.

Estabelecido o problema primal, determina-se o lagrangiano[2] de (1.1), como

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \lambda^T h(x) + \mu^T g(x), \tag{1.2}$$

em que $\mu \in \mathbb{R}^m$ e $\lambda \in \mathbb{R}^l$ são denominadas variáveis duais.

Assim, a partir do lagrangiano, é definido o problema dual de (1.1) da seguinte forma[3]

$$\begin{aligned} \max \quad & q(\mu, \lambda) \\ \text{s.a.} \quad & \mu \geqslant 0 \\ & \lambda \in \mathbb{R}^l, \end{aligned} \tag{1.3}$$

em que $q(\mu, \lambda) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \mu, \lambda)$. Por simplicidade, considera-se que o conjunto de soluções ótimas de (1.3) M^* é não vazio.

Dessa forma, é possível derivar o seguinte resultado.

Proposição 1 (Dualidade Fraca)**.** *Dada as condições anteriores, para qualquer x viável ao problema (1.1) e (λ, μ) viável ao problema (1.3), tem-se que*

$$q(\lambda, \mu) \leqslant f(x).$$

Em particular, seja $f^* = f(x^*)$ e $q^* = q(\mu^*, \lambda^*)$ com $x^* \in X^*$ e $\mu^* \in M^*$, então $q^* \leqslant f^*$.

Demonstração. Considere $\bar{x}$ e $(\bar{\lambda}, \bar{\mu})$ viáveis, ou seja, $g(\bar{x}) \leqslant 0$, $h(\bar{x}) = 0$, $\bar{\mu} \geqslant 0$ e $\bar{\lambda} \in \mathbb{R}^l$. Então, segue que

$$q(\bar{\mu}, \bar{\lambda}) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \bar{\mu}, \bar{\lambda}) \leqslant \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = f(\bar{x}) + \bar{\mu}^T g(\bar{x}) \leqslant f(\bar{x}).$$

[1] Para uma motivação geométrica ver capítulo 4 de Bertsekas (2009).

[2] Apresentado em Bertsekas (2016) S. 3.1.3

[3] No problema dual a variável μ representa o multiplicador de Lagrange relacionado as restrições de desigualdade, por sua vez, este multiplicador possui todas suas coordenadas não negativas, como é discutido em Luenberger e Ye (2016) S. 11.8

Por outro lado, como os vetores ótimos são necessariamente viáveis, vale que

$$q^* \leqslant f^*.$$

■

Por outro lado, é possível garantir que $q^* = f^*$. Para isso, é exigido que as funções f, g e h sejam convexas, de forma que $h(x) = 0$ necessariamente assume a forma afim $Ax - b = 0$. Ademais, será definida a condição de *Slater*[4].

Definição 5. *Seja o problema (1.1) com as funções h e g convexas. A condição de Slater é dita válida se existir um vetor $\bar{x}$ tal que sejam satisfeitas a condição de igualdade e de desigualdade de maneira estrita, ou seja, $A\bar{x} - b = 0$ e $g(\bar{x}) < 0$.*

Entretanto, a priori, serão analisados aspectos pertinentes a matriz $A \in \mathbb{R}^{n \times l}$ que define a restrição de igualdade. A saber, como não há imposições com relação a dependência linear entre as linhas da matriz A, a mesma será denotada, sem perda de generalidade, da seguinte forma

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \tag{1.4}$$

em que as linhas de A_2 são combinações lineares das linhas de A_1 e estas são linearmente independentes. Assim, considere a seguinte proposição.

Proposição 2. *Sejam as funções f, g e h, apresentadas na definição do problema (1.1), convexas, em que $h(x) = Ax - b$. Ainda, considera-se a condição de Slater válida, a matriz A dada por (1.4) e o vetor b dado por $\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ de forma que $Ax = b$. Então os problemas*

$$W = \begin{array}{ll} \min & f(x) \\ \text{s.a.} & Ax - b = 0 \\ & g(x) \leqslant 0 \\ & x \in \mathbb{R}^n \end{array} \qquad Y = \begin{array}{ll} \min & f(x) \\ \text{s.a.} & A_1 x - b_1 = 0 \\ & g(x) \leqslant 0 \\ & x \in \mathbb{R}^n \end{array}$$

possuem os mesmos valores ótimos primal e dual, ou seja, são problemas equivalentes tanto na abordagem primal quanto na dual.

Demonstração. Será denotado, respectivamente, os valores ótimos primal e dual do problema W como f_W^* e q_W^*. Analogamente, para o problema Y, define-se f_Y^* e q_Y^*.

A priori, note que, como a condição de *Slater* é válida, então o sistema linear $Ax - b = 0$ é factível. Assim, é fácil ver que os problemas primais W e Y são

[4] Proposição 3.3.9 de Bertsekas (2016).

equivalentes, ou seja, que $f_W^* = f_Y^*$. De fato, note que, como as linhas de A_2 são combinações lineares das linhas de A_1, então existe uma matriz C tal que

$$\begin{aligned} C^T A_1 &= A_2 \\ C^T b_1 &= b_2. \end{aligned}$$

Logo, o problema W pode ser reescrito da seguinte forma

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & A_1 x - b_1 = 0 \\ & C^T(A_1 x - b_1) = 0 \\ & g(x) \leqslant 0 \\ & x \in \mathbb{R}^n. \end{aligned}$$

Observe que o conjunto de vetores que satisfazem as restrições lineares é dado por

$$\{x \mid A_1 x - b_1 = 0\} \cap \{x \mid A_1 x - b_1 \in Nu(C^T)\} = \{x \mid A_1 x - b_1 = 0\},$$

em que $Nu(C^T)$ representa o núcleo de C^T. Assim, a restrição $C^T(A_1 x - b_1) = 0$ não acrescenta informação ao problema W. Portanto, os problemas primais W e Y são equivalentes, ou seja, $f_W^* = f_Y^*$.

Para o caso dual, note que as funções duais definidas pelos problemas W e Y são dadas por

$$\begin{aligned} q_W(\mu, \lambda_1, \lambda_2) &= \inf_{x \in \mathbb{R}^n} \left\{ f(x) + \mu^T g(x) + \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}^T \left(\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right) \right\} \\ q_Y(\mu, \lambda_1) &= \inf_{x \in \mathbb{R}^n} \{ f(x) + \mu^T g(x) + \lambda_1^T (A_1 x - b_1) \}, \end{aligned}$$

em que a variável dual $\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$. Novamente, utilizando a relação de dependência linear entre A_1 e A_2, tem-se que

$$q_W(\mu, \lambda_1, \lambda_2) = \inf_{x \in \mathbb{R}^n} \{ f(x) + \mu^T g(x) + (\lambda_1 + C\lambda_2)^T (A_1 x - b_1) \}.$$

Por outro lado, observe que o problema dual é definido por (1.3) e considere os vetores $(\mu^W, \lambda_1^W, \lambda_2^W)$ e (μ^Y, λ_1^Y) viáveis, respectivamente, para os problemas W e Y. Então, das expressões de q_W e q_Y apresentadas anteriormente, tem-se que

$$\begin{aligned} & q_W(\mu^W, \lambda_1^W, \lambda_2^W) = q_Y(\mu^W, \lambda_1^W + C\lambda_2^W) \\ \Rightarrow \quad & q_W^* \leqslant q_Y^* \\ & \text{e} \\ & q_Y(\mu^Y, \lambda_1^Y) = q_W(\mu^Y, \lambda_1^Y, 0) \\ \Rightarrow \quad & q_Y^* \leqslant q_W^*. \end{aligned}$$

Portanto, $q_W^* = q_Y^*$, ou seja, os problemas duais de W e Y são equivalentes.

■

Assim, segue o teorema de dualidade forte que garante que $q^* = f^*$.

Teorema 1 (Dualidade Forte). *Seja o problema (1.1) com as funções f, g e h convexas, em que $h(x) = Ax - b$. Ademais, considera-se a condição de Slater válida, então*

$$q^* = f^*.$$

Demonstração. Utilizando a equivalência entre os problemas apresentada na Proposição 2, o problema (1.1) será abordado de forma que a matriz A possua apenas linhas linearmente independentes. Ainda, define-se um conjunto V dado por

$$V = \{(u,v,w) \in \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^l \mid f(x) \leqslant u,\ g(x) \leqslant v \text{ e } Ax - b = w \quad \forall x \in \mathbb{R}^n\}.$$

Note que, uma vez que f e g são convexas e Ax-b linear, então o conjunto V é convexo.

A priori, será demonstrado que $(f^*,0,0)$ não está no interior de V. Por contradição, suponha-se que $(f^*,0,0) \in \text{int}(V)$. Então, existe $\mathcal{E} > 0$ tal que

$$(f^* - \mathcal{E},0,0) \in \text{int}(V),$$

ou seja, $f^* - \mathcal{E}$ é viável para problema (1.1). Mas, $f^* - \mathcal{E} < f^*$ o que contradiz f^* ser mínimo de (1.1). Logo, f^* não pertence ao interior de V.

Assim, pelo teorema do hiperplano de suporte[5], tem-se que existe um vetor $(\mu_0,\mu,\lambda) \in \mathbb{R}^{m+1} - \{0\}$ tal que,

$$\begin{array}{rlll} & (\mu_0,\mu,\lambda)^T(u,v,w) & \geqslant & (\mu_0,\mu,\lambda)^T(f^*,0,0) \quad \forall (u,v,w) \in V \\ \Leftrightarrow & \mu_0 u + \mu^T v + \lambda^T w & \geqslant & \mu_0 f^* \quad \forall (u,v,w) \in V. \end{array} \tag{1.5}$$

Observe que $\mu_0 \geqslant 0$ e $\mu \geqslant 0$. Com efeito, caso $\mu_0 < 0$ ou $\mu_i < 0$ bastaria tomar u ou v_i grande suficiente para tornar a desigualdade (1.5) inconsistente. Dessa forma, será analisado dois casos, a saber

$\mu_0 = 0$

Como o vetor (μ_0,μ,λ) é não nulo, considera-se outros dois casos, a saber

$\mu \neq 0$

Neste caso, de (1.5), tem-se que

$$\begin{array}{rlll} & \mu^T v + \lambda^T w & \geqslant & 0 \quad \forall (u,v,w) \in V \\ \Rightarrow & \inf\limits_{(u,v,w)\in V} \mu^T v + \lambda^T w & \geqslant & 0 \\ \Rightarrow & \inf\limits_x \mu^T g(x) + \lambda^T (Ax - b) & \geqslant & 0, \end{array}$$

uma vez que $w = Ax - b$, $v \geqslant g(x)$ e μ é positivo. Ademais, observe que

$$\inf_{Ax=b} \mu^T g(x) \quad \geqslant \quad \inf_x \mu^T g(x) + \lambda^T (Ax - b).$$

[5] Proposição 1.5.1 de Bertsekas (2009)

Toma-se x restrito a $Ax = b$, pois apenas os pontos viáveis ao problema são de interesse. Por outro lado, utilizando a condição de *Slater*, existe $\bar{x}$ tal que

$$0 \leqslant \inf_{Ax=b} \mu^T g(x) \leqslant \mu^T g(\bar{x}) < 0.$$

Assim, configura-se uma contradição.

$\mu = 0, \lambda \neq 0$

Por outro lado, neste caso, tem-se que

$$\begin{aligned} & \lambda^T w && \geqslant 0 && \forall (u,v,w) \in V \\ \Rightarrow\ & \lambda^T (Ax - b) && \geqslant 0 && x \in \mathbb{R}^n \\ \Rightarrow\ & (A^T\lambda)^T x - \lambda^T b && \geqslant 0 && x \in \mathbb{R}^n. \end{aligned}$$

Note que $A^T\lambda = 0$, pois caso contrario o vetor x poderia ser tomado de forma a não satisfazer essa desigualdade. Logo, esta igualdade contrai o fato das linhas de A serem linearmente independentes, pois $\lambda \neq 0$.

Portanto, μ_0 não pode ser nulo.

$\mu_0 > 0$

A partir de (1.5), tem-se que

$$\begin{aligned} & u + \tilde{\mu}^T v + \tilde{\lambda}^T w && \geqslant f^* && \forall (u,v,w) \in V \\ \Rightarrow\ & \inf_{(u,v,w)\in V} \{u + \tilde{\mu}^T v + \tilde{\lambda}^T w\} && \geqslant f^* \\ \Rightarrow\ & \inf_{x\in\mathbb{R}^n} \{f(x) + \tilde{\mu}^T g(x) + \tilde{\lambda}^T (Ax - b)\} && \geqslant f^* \\ \Rightarrow\ & \sup_{\mu\geqslant 0,\, \lambda\in\mathbb{R}^l} \inf_{x\in\mathbb{R}^n} \{f(x) + \mu^T g(x) + \lambda^T (Ax - b)\} && \geqslant f^*, \end{aligned}$$

em que $(\tilde{\mu}, \tilde{\lambda}) = \left(\dfrac{\mu}{\mu_0}, \dfrac{\lambda}{\mu_0}\right)$. Logo, $q^* \geqslant f^*$.

Dessa forma, a partir da dualidade fraca, é possível afirmar que $q^* = f^*$.

■

Com isso, caso a dualidade forte seja válida, ao determinar o valor ótimo do problema dual trivialmente é resgatado o valor ótimo do problema primal, pois $q^* = f^*$. Ademais, é importante observar que o minimizador do problema primal está, necessariamente, contido no conjunto de minimizadores da função $\mathcal{L}(x, \mu^*, \lambda^*)$, como segue na Proposição 3.

Proposição 3. *Seja a dualidade forte válida e $(x^*, (\mu^*, \lambda^*))$ a solução ótima primal e dual do problema (1.1). Então,*

$$x^* \in \arg\min_{x\in\mathbb{R}^n} \mathcal{L}(x, \mu^*, \lambda^*). \tag{1.6}$$

Demonstração. Se $f^* = q^*$, e $(x^*, (\mu^*, \lambda^*))$ solução dual e primal, então

$$f(x^*) = f^* = q^* = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \mu^*, \lambda^*) \leqslant \mathcal{L}(x^*, \mu^*, \lambda^*).$$

Por outro lado, usando a viabilidade de x^* e o fato de $\mu \geqslant 0$,

$$\mathcal{L}(x^*, \mu^*, \lambda^*) = f(x^*) + g(x^*)^T \mu^* \leqslant f(x^*),$$

logo

$$f(x^*) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \mu^*, \lambda^*) \leqslant \mathcal{L}(x^*, \mu^*, \lambda^*) \leqslant f(x^*).$$

Assim, essa expressão necessariamente é satisfeita por igualdade. Então,

$$\inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \mu^*, \lambda^*) = \mathcal{L}(x^*, \mu^*, \lambda^*).$$

■

1.2.1 *Dualidade de Fenchel*

Considere o problema dado por

$$\begin{array}{ll} \min & f_1(x) + f_2(Ax) \\ \text{s.a.} & x \in \mathbb{R}^n, \end{array} \tag{1.7}$$

em que $f_1 : \mathbb{R}^n \to \mathbb{R}$, $f_2 : \mathbb{R}^m \to \mathbb{R}$ e $A \in \mathbb{R}^{m \times n}$. Pode-se introduzir as variáveis x_1 e x_2 de forma a reescrever o problema como

$$\begin{array}{ll} \min & f_1(x_1) + f_2(x_2) \\ \text{s.a.} & x_2 = Ax_1 \\ & x_1 \in \mathbb{R}^n \\ & x_2 \in \mathbb{R}^m, \end{array} \tag{1.8}$$

Por outro lado, a função dual de (1.8) pode ser reescrita como

$$\begin{aligned} q(\lambda) &= \inf_{(x_1, x_2) \in \mathbb{R}^{m+n}} \{f_1(x_1) + f_2(x_2) + \lambda^T (x_2 - Ax_1)\} \\ &= \inf_{(x_1, x_2) \in \mathbb{R}^{m+n}} \{f_1(x_1) + f_2(x_2) + \lambda^T (x_2 - Ax_1)\} \\ &= \inf_{x_1 \in \mathbb{R}^n} \{f_1(x_1) - x_1^T A^T \lambda\} + \inf_{x_2 \in \mathbb{R}^m} \{f_2(x_2) + x_2^T \lambda\} \\ &= -(f_1^*(A^T \lambda) + f_2^*(-\lambda)), \end{aligned} \tag{1.9}$$

em que foi utilizada a definição de função conjugada (Definição 4). Assim, o problema dual pode ser formulado como

$$\begin{array}{ll} \min & f_1^*(A^T \lambda) + f_2^*(-\lambda) \\ \text{s.a.} & \lambda \in \mathbb{R}^m, \end{array} \tag{1.10}$$

Interpretando o problema dessa forma, é possível, para a classe de problemas definida em (1.7), derivar um caminho alternativo para resgatar o minimizador primal a partir do minimizador dual. A proposição a seguir apresenta tal resultado.

Proposição 4. *Sejam a dualidade forte válida e* (x^*, λ^*) *minimizadores primal* (1.8) *e dual de* (1.10)*, respectivamente. Então,*

$$x^* \in \arg\min_{x\in\mathbb{R}^n}\{f_1(x) - x^T A^T \lambda^*\} \qquad Ax^* \in \arg\min_{z\in\mathbb{R}^m}\{f_2(z) + z^T\lambda^*\}. \tag{1.11}$$

Demonstração. Aplicando a Proposição 3 ao problema (1.8), tem-se que

$$\begin{aligned} & (x_1^*, x_2^*) \in \arg\min_{(x_1,x_2)\in\mathbb{R}^{n+m}} \mathcal{L}(x_1, x_2, \lambda^*) \\ \Rightarrow \quad & (x_1^*, x_2^*) \in \arg\min_{(x_1,x_2)\in\mathbb{R}^{n+m}} \{f_1(x_1) - x_1^T A^T\lambda^* + f_2(x_2) + x_2^T\lambda^*\} \\ \Rightarrow \quad & \begin{array}{l} x_1^* \in \arg\min_{x_1\in\mathbb{R}^n}\{f_1(x_1) - x_1^T A^T\lambda^*\} \\ x_2^* \in \arg\min_{x_2\in\mathbb{R}^m}\{f_2(x_2) + x_2^T\lambda^*\}, \end{array} \end{aligned}$$

utilizando a viabilidade de (x_1^*, x_2^*), toma-se $x_2^* = Ax_1^*$. Assim, denotando $x_1 = x$ e $x_2 = z$, tem-se que

$$\begin{aligned} x^* &\in \arg\min_{x\in\mathbb{R}^n}\{f_1(x) - x^T A^T\lambda^*\} \\ Ax^* &\in \arg\min_{z\in\mathbb{R}^m}\{f_2(z) + z^T\lambda^*\}. \end{aligned}$$

■

1.3 TEORIA DO MÉTODO PROXIMAL

O método proximal procura minimizar irrestritamente uma função convexa não necessariamente diferenciável $f : \mathbb{R}^n \mapsto \mathbb{R}$ a partir de um problema regularizado. Essa minimização é realizada a cada iteração, configurando, assim, um método numérico dado por

Algoritmo 1 – Método Proximal

Dados: x_0 e uma função $c : \mathbb{N} \mapsto \mathbb{R} - \{0\}$
para $l = 0, 1, 2, 3, \ldots$ **faça**
$$x_{l+1} \in \arg\min_{x\in\mathbb{R}^n}\left\{f(x) + \frac{1}{2c_l}||x - x_l||^2\right\}$$
fim

note que o termo de regularização $||x - x_l||^2$ penaliza iterados que se distanciam demais do iterando anterior, ou seja, as novas iterações devem se manter próximas à região já conhecida. Enquanto, o parâmetro c_l é utilizado para controlar o quão próximo da região conhecida o método deve permanecer, ou seja, quanto menor esse parâmetro mais próximo x_{l+1} estará de x_l.

Ademais, a Proposição 5 [Bertsekas (2015)] evidencia a relação de otimalidade intrínseca ao método proximal.

Proposição 5. *Uma sequência $\{x_l\}_{l\in\mathbb{N}}$ é gerada pelo Algoritmo 1, se somente se*

$$\frac{x_l - x_{l+1}}{c_l} \in \partial f(x_{l+1}). \tag{1.12}$$

Demonstração. A partir do problema

$$x_{l+1} \in \arg\min_{x\in\mathbb{R}^n} \left\{ f(x) + \frac{1}{2c_l}||x - x_l||^2 \right\},$$

utilizando o fato da função objetivo ser convexa, é possível aplicar a condição de otimalidade de primeira ordem seguido da soma dos subgradientes[6]. Dessa forma, obtêm-se que

$$0 \in \partial f(x_{l+1}) + \frac{x_{l+1} - x_l}{c_l} \Leftrightarrow \frac{x_l - x_{l+1}}{c_l} \in \partial f(x_{l+1}).$$

■

1.3.1 *Convergência*

A priori, é possível afirmar que o custo da função f decresce a cada iteração, quando aplicado o método proximal. De fato,

$$\begin{aligned} & f(x_{l+1}) + ||x_{l+1} - x_l||^2 \leqslant f(x) + ||x - x_l||^2 \quad \forall x \in \mathbb{R}^n \\ \Rightarrow\ & f(x_{l+1}) + ||x_{l+1} - x_l||^2 \leqslant f(x_l) \\ \Rightarrow\ & f(x_{l+1}) \leqslant f(x_l), \end{aligned} \tag{1.13}$$

ainda, o método proximal garante que a distância a algum minimizador de $f(x)$ também decresce a cada iteração do algoritmo. Esse resultado pode ser derivado diretamente da proposição [Bertsekas (2015)] apresentada a seguir

Proposição 6. *Sejam f convexa e a sequência $\{x_l\}_{l\in\mathbb{N}}$ gerada pelo método proximal. Então, para todo x_l e $c_l > 0$, tem-se que para qualquer $y \in \mathbb{R}^n$*

$$||x_{l+1} - y||^2 \leqslant ||x_l - y||^2 - 2c_l(f(x_{l+1}) - f(y)) - ||x_l - x_{l+1}||^2. \tag{1.14}$$

Demonstração. Seja $y \in \mathbb{R}^n$ qualquer, então

$$\begin{aligned} ||x_l - y||^2 &= ||x_l - x_{l+1} + x_{l+1} - y||^2 \\ &= ||x_l - x_{l+1}||^2 + 2(x_l - x_{l+1})^T(x_{l+1} - y) + ||x_{l+1} - y||^2. \end{aligned}$$

A partir da pertinência (1.12) e da Definição 3, tem-se que

$$\begin{aligned} & f(y) && \geqslant && f(x_{l+1}) + \frac{(x_l - x_{l+1})^T}{c_l}(y - x_{l+1}) \\ \Rightarrow\ & f(x_{l+1}) - f(y) && \leqslant && \frac{(x_l - x_{l+1})^T}{c_l}(x_{l+1} - y) \\ \Rightarrow\ & 2c_l(f(x_{l+1}) - f(y)) && \leqslant && 2(x_l - x_{l+1})^T(x_{l+1} - y), \end{aligned}$$

[6] Proposição 5.4.6 e 5.4.7 de Bertsekas (2009)

substituindo na expressão anterior

$$\begin{aligned} ||x_l - y||^2 &\geqslant ||x_l - x_{l+1}||^2 + 2c_l(f(x_{l+1}) - f(y)) + ||x_{l+1} - y||^2 \\ ||x_{l+1} - y||^2 &\leqslant ||x_l - y||^2 - 2c_l(f(x_{l+1}) - f(y))) - ||x_l - x_{l+1}||^2. \end{aligned}$$

■

Em particular, definindo $f^* = \inf_{x \in \mathbb{R}^n} f(x)$ e $X^* \in \arg\min_{x \in \mathbb{R}^n} f(x)$, obtêm-se que

$$\begin{aligned} ||x_{l+1} - x^*||^2 &\leqslant ||x_l - x^*||^2 - 2c_l(f(x_{l+1} - f^*) - ||x_l - x_{l+1}||^2 \\ ||x_{l+1} - x^*||^2 &\leqslant ||x_l - x^*||^2. \end{aligned} \tag{1.15}$$

Dessa forma, é possível determinar a convergência do método proximal como segue na Proposição 7 [Bertsekas (2015)].

Proposição 7. *Seja a sequência $\{x_l\}$ gerada pelo método proximal (Algoritmo 1). Então, se $\sum_{l=0}^{\infty} c_l = \infty$, tem-se que*

$$f(x_l) \to f^*$$

Ademais, se X^* for não vazio, $\exists\, x^* \in X^*$ tal que

$$x_l \to x^*$$

Demonstração. A priori, define-se f_∞ como

$$f(x_l) \to f_\infty.$$

Como $\{f(x_l)\}$ é não crescente (1.13), então f_∞ está bem definida. Ainda, note que $f_\infty \geqslant f^*$.

Ademais, de (1.14), tem-se que

$$||x_{l+1} - y||^2 \leqslant ||x_l - y||^2 - 2c_l(f(x_{l+1}) - f(y)) \quad y \in \mathbb{R}^n.$$

Dessa forma, somando essa inequação para as $N+1$ primeiras iterações do método, tem-se que

$$\begin{aligned} &||x_1 - y||^2 \leqslant ||x_0 - y||^2 - 2c_0(f(x_1) - f(y)) \\ + \quad &\vdots \\ &||x_{N+1} - y||^2 \leqslant ||x_N - y||^2 - 2c_N(f(x_{N+1}) - f(y)) \end{aligned}$$

$$||x_{N+1} - y||^2 + 2\sum_{l=0}^{N} c_l(f(x_{l+1}) - f(y)) \leqslant ||x_0 - y||^2,$$

ou seja,

$$2\sum_{l=0}^{N} c_l(f(x_{l+1}) - f(y)) \leqslant ||x_0 - y||^2 \quad y \in \mathbb{R}^n, \ N \geqslant 0.$$

Tomando $N \to \infty$, tem-se que

$$2\sum_{l=0}^{\infty} c_l(f(x_{l+1}) - f(y)) \leqslant ||x_0 - y||^2 \quad y \in \mathbb{R}^n. \tag{1.16}$$

Agora, supondo que $f_\infty > f^*$, toma-se $\hat{y}$ tal que

$$f_\infty > f(\hat{y}) > f^*.$$

Dessa forma, como $\{f(x_l)\}$ é não crescente, então

$$f(x_{l+1}) - f(\hat{y}) \geqslant f_\infty - f(\hat{y}) > 0.$$

Logo, de (1.16), é possível derivar

$$2\sum_{l=0}^{\infty} c_l(f(x_{l+1}) - f(\hat{y})) \leqslant ||x_0 - \hat{y}||^2.$$

Assim, usando o fato de $\sum_{l=0}^{\infty} c_l = \infty$, tem-se que

$$\infty \leqslant ||x_0 - y||^2,$$

o que contradiz o fato da norma $||x_0 - y||$ ser finita. Portanto, $f_\infty = f^*$.

Por outro lado, considera-se X^* não vazio e toma-se algum $x^* \in X^*$. Assim, de (1.15), tem-se que $\{x_l\}$ está limitada por uma bola de centro x^* com raio $||x_0 - x^*||$, ou seja,

$$x_l \in \left\{x \mid ||x - x^*||^2 \leqslant ||x_0 - x^*||^2\right\} \quad \forall l.$$

Logo, $\{x_l\}$ pertence a um compacto e, assim, existe ao menos uma subsequência convergente, a saber

$$\lim_{l \in \mathbb{L}} x_l = \bar{x}.$$

Utilizando a Definição 2 de semi-continuidade inferior para f, tem-se que

$$f(\bar{x}) \leqslant \liminf_{l \in \mathbb{L}} f(x_l).$$

Mas, como $f(x_l) \to f^*$, então $f(\bar{x}) \leqslant f^*$. Porém, f^* é mínimo f, logo $f(\bar{x}) = f^*$ e, assim, $\bar{x} \in X^*$. Ou seja, qualquer ponto de acumulação $\{x_l\}$ é minimizador de f.

Agora, será demonstrado que a sequência $\{x_l\}$ possui um único ponto de acumulação. Com efeito, suponha que existam dois pontos de acumulação distintos,

a saber x_1^* e x_2^*. Observe que esses são, necessariamente, minimizadores de f. Assim, existem dois conjuntos de índices dados por $\mathbb{L}_1 \subset \mathbb{N}$ e $\mathbb{L}_2 \subset \mathbb{N}$ tais que

$$\{x_{l_1}\}_{l_1 \in \mathbb{L}_1} \to x_1^* \quad \text{e} \quad \{x_{l_2}\}_{l_2 \in \mathbb{L}_2} \to x_2^*.$$

Logo, existem N_1 e N_2 positivos tais que

$$\begin{aligned} ||x_{l_1} - x_1^*|| &< \frac{1}{2}||x_1^* - x_2^*|| \qquad \forall l_1 \geqslant N_1 \\ ||x_{l_2} - x_2^*|| &< \frac{1}{2}||x_1^* - x_2^*|| \qquad \forall l_2 \geqslant N_2. \end{aligned}$$

Assim, para todo $l_1 \geqslant N_1$ e $l_2 \geqslant N_2$, tem-se que

$$\begin{aligned} & ||x_1^* - x_2^*|| \leqslant ||x_{l_2} - x_1^*|| + ||x_{l_2} - x_2^*|| \\ \Rightarrow \; & ||x_1^* - x_2^*|| < ||x_{l_2} - x_1^*|| + \frac{1}{2}||x_1^* - x_2^*|| \\ \Rightarrow \; & ||x_{l_2} - x_1^*|| > \frac{1}{2}||x_1^* - x_2^*|| > ||x_{l_1} - x_1^*||. \end{aligned}$$

Ou seja, $||x_{l_2} - x_1^*|| > ||x_{l_1} - x_1^*||$. Fato que configura uma contradição com a desigualdade em (1.15). Portanto, $\{x_l\}$ possui apenas um ponto de acumulação.

Por fim, note que a sequência $\{x_l\}$ é limitada com um único ponto de acumulação, logo

$$\{x_l\} \to x^*, \quad x^* \in X^*$$

■

1.3.2 *Proximal Dual*

Nesta subseção, o método proximal será abordado a partir da teoria de dualidade discutida na seção 1.2, em particular a dualidade de *Fenchel*, uma vez que o método proximal pode ser enquadrado na classe de problemas de interesse à dualidade de *Fenchel*. De fato, a função objetivo introduzida pelo método proximal pode ser reescrita como

$$\begin{aligned} & f(x) + \frac{1}{2c_l}||x - x_l||^2 \\ \Leftrightarrow \; & f(x_1) + \frac{1}{2c_l}||x_2 - x_l||^2, \quad x_1 = x_2. \end{aligned}$$

Definindo $f_1(x) = f(x)$ e $f_2(x) = \frac{1}{2c_l}||x - x_l||^2$, obtêm-se um problema na forma

$$\begin{aligned} \min \quad & f_1(x_1) + f_2(x_2) \\ \text{s.a.} \quad & x_2 = x_1 \\ & x_1, x_2 \in \mathbb{R}^n, \end{aligned} \tag{1.17}$$

o qual se enquadra claramente no formato de (1.8).

Portando o problema proximal dual pode ser dado por

$$\begin{array}{ll} \min & f_1^*(\lambda) + f_2^*(-\lambda) \\ \text{s.a.} & \lambda \in \mathbb{R}^n, \end{array} \tag{1.18}$$

em que f_1^* e f_2^* são as funções conjugadas de f_1 e f_2, respectivamente. Agora, buscando uma forma mais conveniente para o problema dual proximal, note que a função conjugada f_2^* é descrita por um problema de maximização quadrática e, portanto, pode ser reescrita da seguinte forma

$$f_2^*(-\lambda) = \sup_{x \in \mathbb{R}^n} \{-\frac{1}{2c_l}||x - x_l||^2 - \lambda^T x\} = -\lambda^T x_l + \frac{c_l}{2}||\lambda||^2.$$

Dessa forma, o problema proximal dual passa a ser dado por

$$\begin{array}{ll} \min & f^*(\lambda) - \lambda^T x_l + \frac{c_l}{2}||\lambda||^2 \\ \text{s.a.} & \lambda \in \mathbb{R}^n, \end{array} \tag{1.19}$$

em que foi utilizado a identidade $f_1 = f$.

Por outro lado, note que a dualidade forte é válida e, portanto, pela Proposição 4 (dualidade de *Fenchel*) é possível resgatar o minimizador primal x_{l+1} a partir do minimizador dual λ_{l+1}, a saber

$$\begin{array}{rcl} x_{l+1} & \in & \arg\max_{x \in \mathbb{R}^n} \{\lambda_{l+1}^T x - f(x)\} \\ x_{l+1} & \in & \arg\max_{x \in \mathbb{R}^n} \{-\lambda_{l+1}^T x - \frac{1}{2c_l}||x - x_l||^2\}. \end{array} \tag{1.20}$$

Utilizando essas relações de pertinência, determina-se as seguintes desigualdades válidas para qualquer $x \in \mathbb{R}^n$

$$\begin{array}{rcl} \lambda_{l+1}^T x - f(x) & \leqslant & \lambda_{l+1}^T x_{l+1} - f(x_{l+1}) \\ (-\lambda_{l+1})^T x - (\frac{1}{2c_l}||x - x_l||^2) & \leqslant & (-\lambda_{l+1})^T x_{l+1} - (\frac{1}{2c_l}||x_{l+1} - x_l||^2). \end{array} \tag{1.21}$$

Assim, é fácil ver que λ_{l+1} e $-\lambda_{l+1}$ satisfazem a Definição 3 de subgradiente para $f(x_{l+1})$ e $\frac{1}{2c_l}||x_{l+1} - x_l||^2$, respectivamente. Ou seja,

$$\lambda_{l+1} \in \partial f(x_{l+1}) \tag{1.22}$$

$$-\lambda_{l+1} \in \partial(\frac{1}{2c_l}||\cdot - x_l||^2)(x_{l+1}). \tag{1.23}$$

Utilizando (1.23) e derivando a função $\frac{1}{2c_l}||x - x_l||^2$, tem-se que

$$\lambda_{l+1} = \frac{x_l - x_{l+1}}{c_l}. \tag{1.24}$$

Por outro lado, substituindo (1.24) em (1.22), obtêm-se que

$$\frac{x_l - x_{l+1}}{c_l} \in \partial f(x_{l+1}). \tag{1.25}$$

Dessa forma, da Proposição 5, é possível inferir que a sequência gerada pelo método proximal dual é em essência a mesma que é gerada pelo método proximal descrito pelo Algoritmo 1. Assim, é possível realizar o método proximal primal (1.17) a partir do dual (1.19). A saber, utilizando (1.24) para determinar x_{l+1}, é possível definir o Algoritmo 2.

Ademais, como as sequências para a variável primal geradas pelo método proximal dual e primal são as mesmas, então, satisfazendo as condições impostas pela Proposição 7, é possível garantir a convergência do método proximal dual.

Algoritmo 2 – Método Proximal Dual

Dados: x_0 e uma função $c : \mathbb{N} \mapsto \mathbb{R} - \{0\}$ tal que $\sum_{l=0}^{\infty} c_l = \infty$

para $l = 0, 1, 2, 3, \ldots$ **faça**

$$\lambda_{l+1} \in \arg\min_{\lambda \in \mathbb{R}^n} \left\{ f^*(\lambda) - \lambda^T x_l + \frac{c_l}{2} ||\lambda||^2 \right\} \tag{1.26}$$

$$x_{l+1} = x_l - c_l \lambda_{l+1} \tag{1.27}$$

fim

1.4 LAGRANGIANO AUMENTADO

O lagrangiano aumentado será derivado apenas para problemas sujeitos a restrições de igualdade[7], uma vez que essa abordagem é suficiente para motivar o método *ADMM*. Dessa forma, considera-se o seguinte problema de otimização

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & Ax - b = 0, \end{aligned} \tag{1.28}$$

em que $f : \mathbb{R}^n \mapsto \mathbb{R}$ é convexa, $A \in \mathbb{R}^{m\,x\,n}$ e $b \in \mathbb{R}^m$.

Definido o problema primal, é introduzida uma grandeza u de forma a perturbar a restrição de igualdade, a saber $Ax - b = u$. Assim, define-se

$$p(u) = \inf_{x \,|\, Ax - b = u} f(x).$$

[7] Na seção 5.2.1 de Bertsekas (2015) é derivado o lagrangiano aumentado para restrições de desigualdade.

Aplicando o método dual ao problema, tem-se que a função dual é dada por

$$\begin{aligned} q(\lambda) &= \inf_{x\in\mathbb{R}^n}\{f(x)+\lambda^T(Ax-b)\} \qquad (1.29)\\ &= \inf_{u\in\mathbb{R}^m}\left\{\inf_{x\,|\,Ax-b=u}\{f(x)+\lambda^T(Ax-b)\}\right\}\\ &= \inf_{u\in\mathbb{R}^m}\left\{\inf_{x\,|\,Ax-b=u}\{f(x)\}+\lambda^T u\right\}\\ &= \inf_{u\in\mathbb{R}^m}\{p(u)+\lambda^T u\}\\ &= -\sup_{u\in\mathbb{R}^m}\{-p(u)-\lambda^T u\}\\ q(\lambda) &= -p^*(-\lambda). \qquad (1.30) \end{aligned}$$

Tomando o problema dual, $\sup_{\lambda\in\mathbb{R}^m} q(\lambda)$, é possível abordá-lo através do método proximal, a saber

$$\lambda_{l+1} \in \arg\max_{\lambda\in\mathbb{R}^m}\left\{q(\lambda)-\frac{1}{2c_l}||\lambda-\lambda_l||^2\right\}, \qquad (1.31)$$

em que $c_l > c > 0$ para todo l com c sendo uma constante positiva, ou seja, $c_l \neq 0$ para qualquer l[8]. Note que ao invés de somar, é subtraído o termo de regularização. Isso se deve ao fato de que a abordagem dual é um problema de maximização. Utilizando a relação em (1.30), tem-se que

$$\begin{aligned} \lambda_{l+1} &\in \arg\max_{\lambda\in\mathbb{R}^m}\left\{-p^*(-\lambda)-\frac{1}{2c_l}||\lambda_l-\lambda||^2\right\}\\ \lambda_{l+1} &\in \arg\min_{\lambda\in\mathbb{R}^m}\left\{p^*(\lambda)+\frac{1}{2c_l}||\lambda_l+\lambda||^2\right\}, \qquad (1.32) \end{aligned}$$

em que foi realizado um abuso de notação, tomando $\lambda = -\lambda$, pois essa mudança não altera o problema. Agora, é possível aplicar o método proximal dual (Algoritmo 2) para resolver (1.32). Utilizando (1.19) e se atentando ao sinal de λ, tem-se que o problema (1.32) passa a ser dado por

$$\inf_{u\in\mathbb{R}^m}\left\{p^{**}(u)+\frac{c_l}{2}||u||^2+u^T\lambda_l\right\}.$$

Ainda, note que $p^{**}(u) = p(u)$, uma vez que p é própria, convexa e semi-contínua inferior[9]. Dessa forma, utilizando a relação (1.22) para determinar λ_{l+1}, o algoritmo proximal dual é dado por

[8] Observe que, dessa forma, ainda é satisfeita a exigência $\sum_{i=0}^{\infty} c_l = \infty$ imposta pelo método proximal.

[9] Proposição 1.6.1 de Bertsekas (2009).

Algoritmo 3 – Método Proximal Dual aplicado ao Problema Dual

Dados: λ_0 e uma função que $c_l : \mathbb{N} \mapsto \mathbb{R}$ tal que $c_l > 0$ para todo l
para $l = 0, 1, 2, 3, \ldots$ **faça**

$$u_{l+1} \in \arg\min_{u \in \mathbb{R}^m} \left\{ p(u) + \frac{c_l}{2}||u||^2 + u^T\lambda_l \right\} \tag{1.33}$$
$$\lambda_{l+1} = \lambda_l + c_l l + 1 \tag{1.34}$$

fim

Assim, de (1.33), tem-se que

$$\begin{aligned}
&\inf_{u \in \mathbb{R}^m} \left\{ \inf_{Ax-b=u} \{f(x)\} + \frac{c_l}{2}||u||^2 + u^T\lambda_l \right\} \\
&\inf_{u \in \mathbb{R}^m} \inf_{Ax-b=u} \left\{ f(x) + \frac{c_l}{2}||Ax-b||^2 + (Ax-b)^T\lambda_l \right\} \\
&\inf_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{c_l}{2}||Ax-b||^2 + (Ax-b)^T\lambda_l \right\} \\
&\inf_{x \in \mathbb{R}^n} \mathcal{L}_{c_l}(x, \lambda_l),
\end{aligned}$$

em que $\mathcal{L}_{c_l}$ é denominado lagrangiano aumentado. Dessa forma, o Algoritmo 3 pode ser reescrito como apresentado no Algoritmo 4.

Algoritmo 4 – Lagrangiano Aumentado

Dados: λ_0 e uma função que $c_l : \mathbb{N} \mapsto \mathbb{R}$ tal que $c_l > 0$ para todo l
para $l = 0, 1, 2, 3, \ldots$ **faça**

$$x_{l+1} \in \inf_{x \in \mathbb{R}^n} \mathcal{L}_{c_l}(x, \lambda_l) \tag{1.35}$$
$$\lambda_{l+1} = \lambda_l + c_l(Ax_{l+1} - b) \tag{1.36}$$

fim

Ademais, a convergência do lagrangiano aumentado pode ser analisada a partir da convergência do método proximal. De fato, como o problema dual é solucionado através do desse método, é possível afirmar que a sequência de variáveis duais $\{\lambda_l\}$ converge para a solução dual desde que o conjunto de soluções ótimas seja não vazio. Assim, de (1.36), é possível derivar que

$$\begin{aligned}
\lambda_{l+1} - \lambda_l &\to 0 \\
c_l(Ax_{l+1} - b) &\to 0.
\end{aligned}$$

Assim, como $c_l > c > 0$ para todo l, então a sequência das variáveis primais $\{x_l\}$ perde inviabilidade com a progressão do método.

Suponha que a sequência $\{x_l\}$ possua ao menos uma subsequência convergente, ou seja, existe $\mathbb{L}$ tal que $\lim_{l\in\mathbb{L}} x_l = \bar{x}$. Dessa forma, tomando o limite inferior em $\mathbb{L}$ e utilizando a definição de semi-continuidade inferior de f, tem-se que

$$f(\bar{x}) \leqslant \liminf_{l\in\mathbb{L}} f(x_l).$$

Por outro lado, como $c_l(Ax_{l+1} - b) \to 0$ para c_l limitado inferiormente, então

$$\liminf_{l\in\mathbb{L}} f(x_l) = \liminf_{l\in\mathbb{L}} \mathcal{L}_{c_{l-1}}(x_l, \lambda_{l-1}).$$

Agora utilizando o fato de x_l minimizar $\mathcal{L}_{c_l}(x, \lambda_{l-1})$, tem-se que

$$\mathcal{L}_{c_{l-1}}(x_l, \lambda_{l-1}) \leqslant \mathcal{L}_{c_{l-1}}(x, \lambda_{l-1}) = f(x) \quad \forall x \mid Ax = b.$$

Logo, $f(\bar{x}) \leqslant f(x)$ para todo x viável, ou seja, $\bar{x} \in X^*$. Portanto, se $\{x_l\}$ possuir ponto de acumulação, este é minimizador de f restrita a $Ax = b$.

1.5 TEORIA DO MÉTODO ADMM

Sejam $f : \mathbb{R}^n \mapsto \mathbb{R}$ e $g : \mathbb{R}^m \mapsto \mathbb{R}$ funções contínuas e convexas, é abordado o seguinte problema

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.a.} \quad & Ax + Bz - d = 0, \end{aligned} \tag{1.37}$$

em que $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{r\,x\,n}$, $B \in \mathbb{R}^{r\,x\,m}$ e $d \in \mathbb{R}^r$. De maneira análoga a utilizada para determinar (1.9), tem-se que o problema dual pode ser descrito como

$$\begin{aligned} \max \quad & v(\lambda) + w(\lambda) \\ \text{s.a.} \quad & \lambda \in \mathbb{R}^r, \end{aligned} \tag{1.38}$$

em que $v(\lambda) = \inf_{x\in\mathbb{R}^n} \{f(x) + \lambda^T Ax\}$ e $w(\lambda) = \inf_{z\in\mathbb{R}^m} \{g(z) + \lambda^T(Bz - d)\}$.

Por outro lado, aplicando o método de lagrangiano Aumentado (Algoritmo 4), tem-se que

$$(x_{l+1}, z_{l+1}) \in \arg\min_{x\in\mathbb{R}^n, z\in\mathbb{R}^m} \{\mathcal{L}_c(x, z, \lambda_l)\} \tag{1.39}$$

$$\lambda_{l+1} = \lambda_l + c(Ax_{l+1} + Bz_{l+1} - d), \tag{1.40}$$

em que $\mathcal{L}_c(x, z, \lambda_l)$ é o lagrangiano Aumentado com c_l constante igual a c, ou seja,

$$\mathcal{L}_c(x, z, \lambda_l) = f(x) + g(z) + \lambda_l^T(Ax + Bz - d) + \frac{c}{2}||Ax + Bz - d||^2. \tag{1.41}$$

Define-se o método das direções alternadas para multiplicadores (*ADMM*) como uma versão radical do método de lagrangiano aumentado inexato, pois é feita apenas uma minimização com respeito a cada variável primal e, em seguida, atualiza-se as variáveis duais. A saber,

Algoritmo 5 – ADMM

Dados: z_0, λ_0 e $c > 0$
para $l = 0, 1, 2, 3, \ldots$ **faça**

$$x_{l+1} \in \arg\min_{x\in\mathbb{R}^n}\{\mathcal{L}_c(x, z_l, \lambda_l)\} \tag{1.42}$$
$$z_{l+1} \in \arg\min_{z\in\mathbb{R}^m}\{\mathcal{L}_c(x_{l+1}, z, \lambda_l)\} \tag{1.43}$$
$$\lambda_{l+1} = \lambda_l + c(Ax_{l+1} + Bz_{l+1} - d) \tag{1.44}$$

fim

1.5.1 *Convergência*

Nesta subseção será tratada a convergência[10] do método *ADMM* para o problema apresentado em (1.37).

Proposição 8. *Seja o problema primal dado por* (1.37) *com dual dado por* (1.38)*, em que* X^* *e* M^* *são não vazios. Então, todo ponto de acumulação* $(\bar{x}, \bar{z}, \bar{\lambda})$ *da sequência gerada pelo método ADMM é tal que* $(\bar{x}, \bar{z})$ *é minimizador do problema primal e* $\bar{\lambda}$ *é minimizador do problema dual.*

Demonstração. Inicialmente, toma-se o problema (1.8) com $A = I$. Assim, utilizando a dualidade de *Fenchel* (Proposição 4) tem-se que

$$x_{l+1} \in \arg\max_{x\in\mathbb{R}^n}(\theta_{l+1}^T x - f_1(x))$$
$$x_{l+1} \in \arg\max_{x\in\mathbb{R}^n}(-\theta_{l+1}^T x - f_2(x)),$$

em que θ é a variável dual de (1.10). Aplicando a condição de otimalidade, tem-se que

$$\theta_{l+1} \in \partial f_1(x_{l+1}) \tag{1.45}$$
$$-\theta_{l+1} \in \partial f_2(x_{l+1}). \tag{1.46}$$

Dessa forma, observe que

$$x_{l+1} \in \arg\min_{x\in\mathbb{R}^n}\{f_2(x) + g_{l+1}^T x\}, \tag{1.47}$$

com $g_{l+1} \in \partial f_1(x_{l+1})$.

Agora, considera-se o primeiro subproblema do *ADMM*, apresentado em (1.42), junto com as seguintes atribuições

$$\begin{aligned} f_2(x) &= f(x) \\ f_1(x) &= \lambda_l^T Ax + \frac{c}{2}||Ax + Bz_l - d||^2. \end{aligned} \tag{1.48}$$

[10] A demonstração apresentada é uma adaptação da proposição 4.2 de Bertsekas e Tsitsikhs (1997).

Assim, utilizando a pertinência em (1.47) e notando que

$$g_{l+1} = A^T\lambda_l + cA^T(Ax_{l+1} + Bz_l - d), \tag{1.49}$$

tem-se que

$$\begin{aligned} &f(x_{l+1}) + [\lambda_l + c(Ax_{l+1} + Bz_l - d)]^T Ax_{l+1} \\ &\quad \leqslant f(x) + [\lambda_l + c(Ax_{l+1} + Bz_l - d)]^T Ax \end{aligned} \tag{1.50}$$

para todo $x \in \mathbb{R}^n$. Analogamente, é possível resgatar o problema com relação a variável primal z (1.43). Para isso, realiza-se as seguintes atribuições ao problema (1.47) em função da variável z

$$\begin{aligned} f_2(z) &= g(z) \\ f_1(z) &= \lambda_l^T Bz + \frac{c}{2}||Ax_{l+1} + Bz - d||^2. \end{aligned} \tag{1.51}$$

Novamente da pertinência em (1.47) e notando que

$$g_{l+1} = B^T\lambda_l + cB^T(Ax_{l+1} + Bz_{l+1} - d), \tag{1.52}$$

tem-se que

$$\begin{aligned} &g(z_{l+1}) + [\lambda_l + c(Ax_{l+1} + Bz_{l+1} - d)]^T Bz_{l+1} \\ &\quad \leqslant g(z) + [\lambda_l + c(Ax_{l+1} + Bz_{l+1} - d)]^T Bz \end{aligned} \tag{1.53}$$

para todo $z \in \mathbb{R}^m$.

Note que, a partir da iteração para a variável dual λ (1.44), tem-se que

$$\lambda_l = \lambda_{l+1} - c(Ax_{l+1} + Bz_{l+1} - d). \tag{1.54}$$

Assim, substitui-se (1.54) em (1.50) e (1.53), de forma a obter as seguintes desigualdades válidas para todo (x, z)

$$\begin{aligned} &f(x_{l+1}) + \lambda_{l+1}^T Ax_{l+1} + c(Bz_l - Bz_{l+1})^T Ax_{l+1} \\ &\quad \leqslant f(x) + \lambda_{l+1}^T Ax + c(Bz_l - Bz_{l+1})^T Ax \\ &g(z_{l+1}) + \lambda_{l+1}^T Bz_{l+1} \\ &\quad \leqslant g(z) + \lambda_{l+1}^T Bz. \end{aligned} \tag{1.55}$$

Considerando que X^* é não vazio, então, em particular, essas desigualdades são válidas para minimizadores primais, a saber

$$\begin{aligned} &f(x_{l+1}) + \lambda_{l+1}^T Ax_{l+1} + c(Bz_l - Bz_{l+1})^T Ax_{l+1} \\ &\quad \leqslant f(x^*) + \lambda_{l+1}^T Ax^* + c(Bz_l - Bz_{l+1})^T Ax^* \\ &g(z_{l+1}) + \lambda_{l+1}^T Bz_{l+1} \\ &\quad \leqslant g(z^*) + \lambda_{l+1}^T Bz^*, \end{aligned} \tag{1.56}$$

em que (x^*, z^*) é uma solução do problema primal. Dessa forma, somando essas duas desigualdades e usando o fato de $Ax^* + Bz^* = d$, então

$$\begin{aligned} f(x_{l+1}) + g(z_{l+1}) + \lambda_{l+1}^T(Ax_{l+1} + Bz_{l+1} - d) \\ + c(Bz_l - Bz_{l+1})^T A(x_{l+1} - x^*) \leqslant f(x^*) + g(z^*). \end{aligned} \tag{1.57}$$

Pela Proposição 3, para todo (x,z) e toda solução dual $\lambda^* \in M^*$, tem-se que

$$\begin{aligned} & \mathcal{L}(x^*,z^*,\lambda^*) = \inf_{x\in\mathbb{R}^n, z\in\mathbb{R}^m} \mathcal{L}(x,z,\lambda^*) \leqslant \mathcal{L}(x,z,\lambda^*) \\ \Rightarrow \quad & f(x^*) + g(z^*) \leqslant f(x) + g(z) + \lambda^*(Ax + Bz - d). \end{aligned} \tag{1.58}$$

Somando as desigualdades (1.57) e (1.58) computada em (x_{l+1}, z_{l+1}), conclui-se que

$$(\lambda_{l+1} - \lambda^*)^T(Ax_{l+1} + Bz_{l+1} - d) + c(Bz_l - Bz_{l+1})^T A(x_{l+1} - x^*) \leqslant 0. \tag{1.59}$$

Por outro lado, define-se as seguintes variáveis

$$\bar{x}_l = x_l - x^* \qquad \bar{z}_l = z_l - z^* \qquad \bar{\lambda}_l = \lambda_l - \lambda^*. \tag{1.60}$$

Assim, a expressão (1.44) pode ser reescrita de duas formas distintas utilizando as novas variáveis, a saber

$$Ax_{l+1} + Bz_{l+1} - d = \frac{1}{c}(\bar{\lambda}_{l+1} - \bar{\lambda}_l) \tag{1.61}$$

$$A\bar{x}_{l+1} = \frac{1}{c}(\bar{\lambda}_{l+1} - \bar{\lambda}_l) - B\bar{z}_{l+1}, \tag{1.62}$$

em que a última expressão faz uso da igualdade $Ax^* + Bz^* = d$. Dessa forma, utilizando as novas variáveis e as identidades (1.61) e (1.62), a desigualdade (1.59) pode ser reescrita como

$$\frac{1}{c}\bar{\lambda}_{l+1}^T(\bar{\lambda}_{l+1} - \bar{\lambda}_l) + (B\bar{z}_l - B\bar{z}_{l+1})^T(\bar{\lambda}_{l+1} - \bar{\lambda}_l) - c(B\bar{z}_l - B\bar{z}_{l+1})^T B\bar{z}_{l+1} \leqslant 0. \tag{1.63}$$

Ainda, note que as seguintes identidades são válidas

$$\begin{aligned} \bar{\lambda}_{l+1}^T(\bar{\lambda}_{l+1} - \bar{\lambda}_l) &= \frac{1}{2}||\bar{\lambda}_{l+1} - \bar{\lambda}_l||^2 + \frac{1}{2}||\bar{\lambda}_{l+1}||^2 - \frac{1}{2}||\bar{\lambda}_l||^2 \\ (B\bar{z}_l - B\bar{z}_{l+1})^T B\bar{z}_{l+1} &= -\left(\frac{1}{2}||B\bar{z}_{l+1} - B\bar{z}_l||^2 + \frac{1}{2}||B\bar{z}_{l+1}||^2 - \frac{1}{2}||B\bar{z}_l||^2\right). \end{aligned} \tag{1.64}$$

Por outro lado, ainda é possível mostrar que $(B\bar{z}_l - B\bar{z}_{l+1})^T(\bar{\lambda}_{l+1} - \bar{\lambda}_l)$ é não negativo. De fato, da segunda desigualdade de (1.55) computada em z_{l+1}, tem-se que

$$g(z_{l+1}) + \lambda_{l+1}^T Bz_{l+1} \leqslant g(z_l) + \lambda_{l+1}^T Bz_l.$$

Ainda utilizando (1.55) computada em z_{l+1}, porém agora tomando a iteração l ao invés da iteração $l+1$, é possível derivar a seguinte desigualdade

$$g(z_l) + \lambda_l^T Bz_l \leqslant g(z_{l+1}) + \lambda_l^T Bz_{l+1}.$$

Somando essas duas últimas desigualdades e transformando as variáveis, obtêm-se o resultado esperado

$$0 \leqslant (B\bar{z}_l - B\bar{z}_{l+1})^T(\bar{\lambda}_{l+1} - \bar{\lambda}_l). \tag{1.65}$$

Assim, aplicando as desigualdades (1.64) e (1.65) em (1.63), tem-se que

$$\begin{aligned} 0 &\leqslant ||\bar{\lambda}_{l+1} - \bar{\lambda}_l||^2 + c^2||B\bar{z}_{l+1} - B\bar{z}_l||^2 \\ &\leqslant (||\bar{\lambda}_l||^2 + c^2||B\bar{z}_l||^2) - (||\bar{\lambda}_{l+1}||^2 + c^2||B\bar{z}_{l+1}||^2), \end{aligned} \tag{1.66}$$

desigualdade que implica em

$$\begin{aligned} \bar{\lambda}_{l+1} - \bar{\lambda}_l &\to 0 \\ B\bar{z}_{l+1} - B\bar{z}_l &\to 0. \end{aligned} \tag{1.67}$$

Com efeito, define-se

$$\begin{aligned} \alpha_l &= ||\bar{\lambda}_l||^2 + c^2||B\bar{z}_l||^2 \\ \gamma_l &= ||\bar{\lambda}_{l_{j+1}} - \bar{\lambda}_{l_j}||^2 + c^2||B\bar{z}_{l_{j+1}} - B\bar{z}_{l_j}||^2. \end{aligned}$$

Inicialmente, note que para mostrar (1.67), basta concluir que $\gamma_l \to 0$. Para isso, a partir de (1.66), facilmente é possível concluir que $\{\alpha_l\}$ é uma sequência não crescente e limitada inferiormente por 0, portanto, $\{\alpha_l\}$ converge, em particular $\alpha_l - \alpha_{l+1} \to 0$. Assim, tomando o limite sobre a desigualdade (1.66), tem-se que

$$0 \leqslant \lim_{l \in \infty} \gamma_l \leqslant 0,$$

logo $\gamma_l \to 0$.

Agora, suponha que a sequência $\{(x_{l+1}, z_{l+1}, \lambda_{l+1})\}$ gerada pelo método *ADMM* possua ao menos um ponto de acumulação, a saber

$$\lim_{l \in \mathbb{L}} (x_{l+1}, z_{l+1}, \lambda_{l+1}) = (\tilde{x}, \tilde{y}, \tilde{\lambda}).$$

Dessa forma, a partir da expressão (1.61), é possível afirmar que $\tilde{x}$ e $\tilde{z}$ são viáveis, ou seja,

$$A\tilde{x} + B\tilde{z} = d.$$

Assim, tomando o limite para $l \in \mathbb{L}$ nas desigualdades (1.57) e (1.58) (esta última computada em $l + 1$), uma vez que f e g são contínuas, tem-se que

$$\begin{aligned} & f(x^*) + g(z^*) \geqslant f(\tilde{x}) + g(\tilde{z}) \\ & f(x^*) + g(z^*) \leqslant f(\tilde{x}) + g(\tilde{z}) \\ \Rightarrow \quad & f(x^*) + g(z^*) = f(\tilde{x}) + g(\tilde{z}). \end{aligned} \tag{1.68}$$

Logo, os pontos de acumulação $\tilde{x}$ e $\tilde{z}$ são soluções primais.

Por outro lado, ainda resta mostrar que $\tilde{\lambda}$ é solução dual. Para isso, inicialmente define-se a seguinte variável

$$\hat{\lambda}_{l+1} = \lambda_l + c(Ax_{l+1} + Bz_l - d). \tag{1.69}$$

Note que o subgradiente derivado em (1.49) pode ser reescrito em função da variável $\hat{\lambda}$, possibilitando reescrever a pertinência em (1.47) como

$$x_{l+1} \in \arg\min_{x\in\mathbb{R}^n}\{f(x) + \hat{\lambda}_{l+1}^T Ax\}. \tag{1.70}$$

Asim, a partir da definição da função $v(\lambda)$ em (1.38), tem-se que

$$v(\hat{\lambda}_{l+1}) = f(x_{l+1}) + \hat{\lambda}_{l+1}^T Ax_{l+1}. \tag{1.71}$$

Por outro lado, com as atribuições em (1.51) e utilizando a iteração para a variável dual em (1.44), a pertinência (1.47) é dada por

$$\begin{aligned} z_{l+1} &\in \arg\min_{z\in\mathbb{R}^m}\{g(z) + \lambda_{l+1}^T Bz\} \\ \Rightarrow\ z_{l+1} &\in \arg\min_{z\in\mathbb{R}^m}\{g(z) + \lambda_{l+1}^T Bz + \lambda_{l+1}^T d\}. \end{aligned} \tag{1.72}$$

Dessa forma, utilizando a definição da função $w(\lambda)$ em (1.38), é possível derivar que

$$w(\lambda_{l+1}) = g(z_{l+1}) + \lambda_{l+1}^T Bz_{l+1} - \lambda_{l+1}^T d \tag{1.73}$$

Agora, somando (1.71) e (1.73), utilizando a definição de $\hat{\lambda}$ e a iteração segundo a variável dual em (1.44), tem-se que

$$\begin{aligned} w(\lambda_{l+1}) + v(\hat{\lambda}_{l+1}) = f(x_{l+1}) + g(z_{l+1}) + \lambda_l^T(Ax_{l+1} + Bz_{l+1} - d) + \\ c(Ax_{l+1} + Bz_{l+1} - d)^T(Bz_{l+1} - d) + c(Ax_{l+1} + Bz_{l+1} - d)^T Ax_{l+1}. \end{aligned} \tag{1.74}$$

Tomando o limite de $l \in \mathbb{L}$, tem-se que

$$\lim_{l\in\mathbb{L}}(v(\hat{\lambda}_{l+1}) + w(\lambda_{l+1})) = f(x^*) + g(z^*), \tag{1.75}$$

em que foi utilizada a viabilidade de $(\tilde{x},\tilde{z})$ e o resultado apresentado em (1.67). Mas, pela dualidade forte, é válido que

$$\sup_{\lambda\in\mathbb{R}^l}\{v(\lambda) + w(\lambda)\} = \inf_{Ax+Bz-d=0}\{f(x) + g(z)\} = f(x^*) + g(z^*).$$

Logo,

$$\lim_{l\in\mathbb{L}}(v(\hat{\lambda}_{l+1}) + w(\lambda_{l+1})) = \sup_{\lambda\in\mathbb{R}^l}\{v(\lambda) + w(\lambda)\}. \tag{1.76}$$

Por fim, observando que, pela definição de $\hat{\lambda}$ alinhada ao resultado (1.67), $\lim_{l\in\mathbb{L}}\hat{\lambda}_{l+1} = \tilde{\lambda}$ e utilizando a semi-continuidade superior de v e w, tem-se que

$$\begin{aligned} v(\tilde{\lambda}) + w(\tilde{\lambda}) &\geqslant \sup_{\lambda\in\mathbb{R}^l}(v(\lambda) + w(\lambda)) \\ \Rightarrow\ v(\tilde{\lambda}) + w(\tilde{\lambda}) &= \sup_{\lambda\in\mathbb{R}^l}(v(\lambda) + w(\lambda)), \end{aligned} \tag{1.77}$$

Portanto, $\tilde{\lambda}$ é solução dual, ou seja, $\tilde{\lambda} \in M^*$.

■

CAPÍTULO 2

Máquina de Suporte Vetorial

O paradigma de máquina de suporte vetorial SVM é largamente utilizado no contexto de aprendizado de máquina para problemas supervisionados. O SVM será explorado tanto em sua formulação centralizada como, principalmente, em sua formulação distribuída. Utilizando, para isso, o método ADMM.

2.1 Motivação

Tradicionalmente, o *SVM* é introduzido como um paradigma voltado ao aprendizado supervisionado em classificação binária, ou seja, cria um modelo preditivo a partir de um conjunto de dados previamente conhecido que relaciona um determinado vetor $x \in \mathbb{R}^n$ a uma de duas classes quaisquer, a saber A e B. Usualmente, o conjunto de dados é denominado por conjunto de treino, enquanto cada coordenada x_i do vetor x é denominada por característica ou atributo.

Para criar esse modelo preditivo, o *SVM* busca determinar o hiperplano que melhor separa as classes A e B de forma a servir como uma fronteira de mudança de classe utilizada para a predição de dados futuros. Essa busca por conceituar o que vem

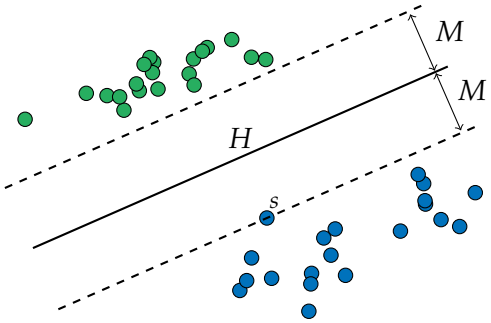


Figura 1 – Aspecto geral de um modelo *SVM*, no qual determina o hiperplano H que melhor separa as classes (dados em azul e em vermelho). M representa a margem, enquanto o dado rotulado por s é denominado vetor de suporte.

a ser o melhor hiperplano de separação é o que leva à ideia central do *SVM*, a saber, o conceito de margem. A ideia de margem se deve à intuição de que quanto mais distante a fronteira de separação estiver dos dados rotulados por A e por B, então mais

confiável o modelo aparenta ser, ou seja, menos suscetível a erros de predição. Portanto, o hiperplano é construído de tal forma a maximizar a menor distância relativa a este hiperplano tanto entre os dados de classe A quanto entre os dados de classe B. A distância que decorre dessa maximização é denominada por margem do conjunto de treino.

2.2 CLASSIFICAÇÃO LINEAR CENTRALIZADA

Nesta seção será analisado o problema de suporte de máquina vetorial para um conjunto de dados S definido por

$$S = \{(x_i, y_i) \mid x_i \in \mathbb{R}^n,\ y_i \in \{-1, 1\} \text{ e } i = 1, \cdots, m\}, \tag{2.1}$$

o qual é tipicamente denominado por conjunto de treino. Considera-se que os dados desse conjunto possam ser satisfatoriamente separados de acordo com as duas classes 1 e -1 por um hiperplano.

2.2.1 *Problema Primal para a Margem Rígida*

Inicialmente, considera-se que os dados do conjunto de treino S podem ser separados não apenas de forma satisfatória, mas de forma exata por um hiperplano. O qual é definido por um vetor $(w, b) \in \mathbb{R}^{n+1}$, a saber $H = \{x \,|\, (w, 1)^T (x, b) = 0\}$.

Para formalizar o conceito de margem, serão abordados aspectos geométricos concernentes ao problema de suporte de máquina vetorial. Com isso, toma-se um dado (x, y) qualquer do conjunto S. Definido um ponto x, é possível determinar a distância relativa desse ponto ao hiperplano a partir de sua projeção ao mesmo, como apresentado na Figura 2.

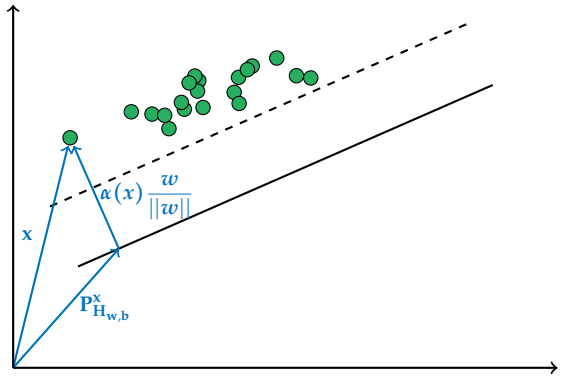


Figura 2 – Noção geométrica para a distância relativa de um dado $(x, y) \in S$.

Note que a projeção $P^x_{H_{w,b}}$ pertence ao hiperplano H, ou seja,

$$w^T P^x_{H_{w,b}} + b = 0. \tag{2.2}$$

Ainda, observe que $x - \mathrm{P}^x_{H_{w,b}}$ é ortogonal a H e, assim, paralelo a w, ou seja,

$$x - \mathrm{P}^x_{H_{w,b}} = \alpha(x)\frac{w}{||w||}, \tag{2.3}$$

em que $\alpha(x) \in \mathbb{R}$, note que a orientação do vetor w determina o sinal de $\alpha(x)$. Dessa forma, utilizando (2.2) e (2.3), é possível concluir que

$$\alpha(x) = \frac{w^T}{||w||}x + \frac{b}{||w||}. \tag{2.4}$$

Agora, buscando uma grandeza absoluta de forma a mensurar uma distância, define-se os dados rotulados por $y = 1$ aqueles que possuem o vetor $x - P^x_{H_{w,b}}$ com mesmo sentido que o vetor w enquanto $y = -1$ caracteriza os dados que possuem esse vetor com sentido oposto ao vetor w. Dessa forma, define-se

$$f_m(x,y,w,b) = y\left(\frac{w^T}{||w||}x + \frac{b}{||w||}\right), \tag{2.5}$$

em que f_m é denominada função margem, a qual determina a distância relativa entre um dado $(x,y) \in S$ e o hiperplano H definido um vetor $(w,b) \in \mathbb{R}^{n+1}$. Ademais, é importante ressaltar que a função margem é invariante por múltiplos escalares do vetor (w,b) devido ao fator $\dfrac{1}{||w||}$.

Além disso, a função margem pode ser utilizada para averiguar se a classificação de um dado (x,y) frente a um determinado hiperplano está correta ou não. De fato, basta notar que a classe obtida para um ponto x é dada por

$$\operatorname{sign}\left(\frac{w^T}{||w||}x + \frac{b}{||w||}\right). \tag{2.6}$$

Logo, se essa classificação corresponder a classe y, ou seja, se a classificação estiver correta, então $f_m(x,y,w,b) > 0$. Caso contrário $f_m(x,y,w,b) < 0$. Comumente essa avaliação é realizada frente a um conjunto disjunto de S denominado conjunto de teste, o qual é previamente conhecido. Pois dessa forma, é possível estimar a acurácia determinada pelo classificador. Naturalmente, este é posteriormente aplicado a novos dados que não possuem uma classe associada.

Por fim, determinada a função margem, é possível definir a margem de um dado hiperplano H como sendo

$$M(w,b) = \min_{(x,y)\in S} f_m(x,y,w,b). \tag{2.7}$$

em que os vetores $(x,y) \in \arg\min_{(x,y)\in S} f_m(x,y,w,b)$ são denominados vetores de suporte para o hiperplano H. Ainda, é possível definir a margem de um conjunto de treino S como

$$M = \sup_{(w,b)\in\mathbb{R}^{n+1}} M(w,b). \tag{2.8}$$

Contudo, esse problema pode ser reescrito como

$$\begin{aligned} \max \quad & M(w,b) \\ \text{s.a.} \quad & M(w,b) - y_i \left(\frac{w^T}{||w||} x_i + \frac{b}{||w||} \right) \leqslant 0 \quad i = 1, \cdots, m \\ & (w,b) \in \mathbb{R}^{n+1}. \end{aligned} \tag{2.9}$$

em que a restrição de desigualdade, inicialmente, não acrescenta informação ao problema, pois é redundante se comparada a definição de $M(w,b)$. Agora, utilizando o fato da função margem ser invariante por múltiplos escalares de (w,b), toma-se uma escala tal que

$$||w|| = \frac{1}{M_{w,b}}. \tag{2.10}$$

Assim o problema (2.9) pode ser reescrito como

$$\begin{aligned} \max \quad & \frac{1}{||w||} \\ \text{s.a.} \quad & 1 - y_i \left(w^T x_i + b \right) \leqslant 0 \quad i = 1, \cdots, m \\ & (w,b) \in \mathbb{R}^{n+1}. \end{aligned} \tag{2.11}$$

Ainda, como $||w||$ é necessariamente positiva, então é possível afirmar que maximizar a razão $\frac{1}{||w||}$ é o mesmo que minimizar $||w||$ ou, equivalentemente, $\frac{1}{2}||w||^2$. Portanto, o problema (2.11) pode ser reescrito como um problema convexo, a saber

$$\begin{aligned} \min \quad & \frac{1}{2}||w||^2 \\ \text{s.a.} \quad & 1 - y_i(w^T x_i + b) \leqslant 0 \quad i = 1, \cdots, m \\ & (w,b) \in \mathbb{R}^{n+1}. \end{aligned} \tag{2.12}$$

2.2.2 *Conjunto de Dados não Separáveis*

Diferente da seção anterior, agora o problema de suporte vetorial será analisado para os casos em que os dados de S não podem ser separados de forma exata por um hiperplano, porém ainda são linearmente separados de forma satisfatória. Ou seja, será abordada a situação em que aplicar o problema de margem rígida ao conjunto de treino é impraticável. Ainda, note que seria possível considerar o caso em que os dados não podem ser linearmente separados pois estes possuem comportamento não linear, porém esta situação será discutida posteriormente.

Dado um hiperplano H definido a partir de um vetor $(w,b) \in \mathbb{R}^{n+1}$, como os dados em S não podem ser separados linearmente de forma exata, então é natural existirem dados de treino que estão erroneamente classificados por H. Assim, define-se como ε_i a distância relativa entre H e um dado (x_i, y_i) classificado de forma equivocada, ou seja, ε_i quantifica o erro associado a um dado (x_i, y_i), assim como apresentado na Figura 3. Observe que ε é nulo para os dados que estão corretamente classificados.

Para esse caso, a ideia central é permitir que hiperplanos que comportam um número pequeno de erros de classificação frente ao conjunto de treino sejam factíveis ao problema de *SVM*, por exemplo, no caso da Figura 3 há três dados com classificados de forma equivocada. Para tanto, é necessário lidar com dois conceitos conflitantes, a saber, a flexibilização a margem e a limitação do erro.

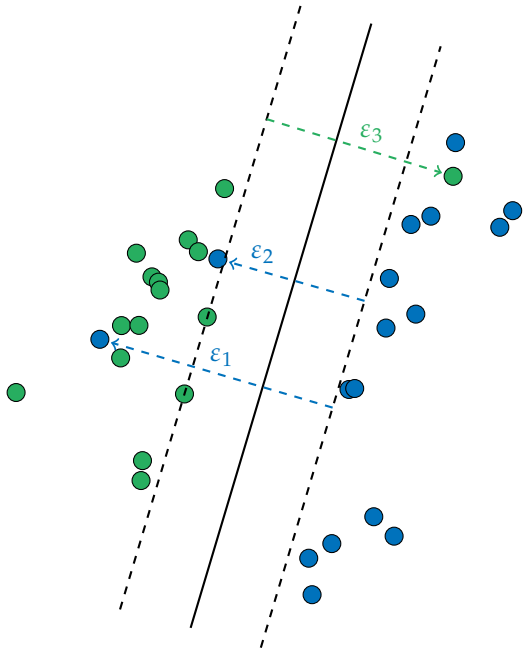


Figura 3 – ε_1, ε_2 e ε_3 são os erros relativos a um hiperplano apresentados por dados classificados erroneamente pelo mesmo.

Com o intuito de flexibilizar a margem é possível alterar[1] a restrição de desigualdade derivada em (2.9), a saber

$$y_i(w^T x_i + b) \quad \geqslant \quad M(w,b)(1-\varepsilon_i),$$

em que a margem para um dado hiperplano é proporcionalmente flexibilizada pelo tamanho do erro ε_i de forma a tornar o hiperplano definido por (w,b) viável ao problema.

Por outro lado, para limitar o erro, será utilizado um termo de regularização de norma 1 acrescido a função objetivo[2]. Assim, o problema (2.12) pode ser reescrito

[1] Seção 12.2 de Hastie, Tibshirani e Friedman (2013).

[2] A norma 1 é utilizada com o intuito de impor ao método que cometa menos erros. Pois, para para qualquer $x \in \mathbb{R}^n - \{0\}$, vale que

$$||x||_1 > ||x||_q, \quad \forall q \geqslant 2.$$

como

$$\begin{aligned} \min \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\varepsilon_i \\ \text{s.a.} \quad & 1-\varepsilon_i - y_i(w^T x_i + b) \leqslant 0 \quad i = 1,\cdots,m \\ & -\varepsilon_i \leqslant 0 \qquad\qquad\qquad\quad i = 1,\cdots,m \\ & (w,b) \in \mathbb{R}^{n+1}, \end{aligned} \tag{2.13}$$

em que o parâmetro C pondera o quão flexível o método será com relação aos erros, ou seja, quanto maior C menos permissivo à erros intrínsecos o método será.

2.2.3 *Abordagem Clássica*

Claramente há diversas maneiraras em que o problema (2.13) pode ser abordado, contudo a abordagem dual para solucionar esse problema é laragamente empregada[3]. Por isso, serão apresentados alguns detalhes dessa abordagem de forma a motivar o uso de núcleos para o caso da classificação não linear.

Inicialmente, note que a condição de *Slater* é satisfeita para o problema (2.13). Portanto, o mesmo pode ser abordado através do problema dual, a saber

$$\begin{aligned} \max \quad & q(\mu,\eta) \\ \text{s.a.} \quad & \mu \geqslant 0 \\ & \eta \geqslant 0 \end{aligned} \tag{2.14}$$

em que $q(\mu,\eta) = \inf_{(w,b,\varepsilon)\in\mathbb{R}^{n+2}} \mathcal{L}(w,b,\varepsilon,\mu,\eta)$ com o lagrangeano dado por

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\varepsilon_i - \sum_{i=1}^{m}\mu_i(\varepsilon_i + y_i(w^T x_i + b) - 1) - \sum_{i=1}^{m}\eta_i\varepsilon_i. \tag{2.15}$$

Ademais, note que, como o lagrangeano é uma função quadrática, é possível determinar uma forma fechada para a função dual. Para isso, toma-se a condição de primeira ordem

$$w - \sum_{i=1}^{m}\mu_i y_i x_i = 0 \tag{2.16}$$

$$\sum_{i=1}^{m}\mu_i y_i = 0 \tag{2.17}$$

$$C - \mu_i - \eta_i = 0 \quad i = 1,\cdots,m. \tag{2.18}$$

Assim, substituindo (2.16), (2.17) e (2.18) no lagrangeano, é possível reescrever a função dual como

$$q(\mu) = \sum_{i=1}^{m}\mu_i - \sum_{i=1}^{m}\sum_{j=1}^{m}\mu_i\mu_j y_i y_j x_i^T x_j. \tag{2.19}$$

[3] Subseção 12.2.1 de Hastie, Tibshirani e Friedman (2013)

Ainda, de (2.17), (2.18) e observando que $\eta \geqslant 0$, o problema dual pode ser reescrito como

$$\begin{aligned} \max \quad & q(\mu) \\ \text{s.a.} \quad & \sum_{i=1}^{m} \mu_i y_i = 0 \\ & 0 \leqslant \mu_i \leqslant C \quad i = 1, \cdots, m. \end{aligned} \tag{2.20}$$

Por outro lado, é importante notar que, a partir de (2.16), tem-se que

$$w^* = \sum_{i=1}^{m} \mu_i^* y_i x_i. \tag{2.21}$$

Logo, w^* pode ser reescrito apenas pelos vetores x_i tais que satisfazem a restrição

$$1 - \varepsilon_i - y_i(w^T x_i + b) \leqslant 0 \tag{2.22}$$

por igualdade. Pois, apenas para esses vetores a variável dual μ_i não é necessariamente nula[4]. Devido a sua importância, esses vetores são denominados por vetores de suporte, os quais são ilustrados na Figura 1.

Ainda de (2.21), é possível derivar que a classificação de um vetor x pode ser realizada apenas utilizando as variáveis duais μ e o escalar b, a saber

$$\operatorname{sign}(w^T x + b) = \operatorname{sign}\left(\sum_{i=1}^{m} \mu_i y_i x_i^T x + b \right), \tag{2.23}$$

Por fim, observe que, uma vez determinada as variáveis duais, é possível determinar b através dos vetores de suporte. Pois, para qualquer vetor de suporte $\varepsilon = 0$ e, como já discutido, a restrição (2.22) é ativa, ou seja, para qualquer vetor de suporte x_i, tem-se que

$$1 - y_i(w^T x_i + b) = 0 \tag{2.24}$$

Enfim, o problema (2.20) pode ser solucionado, por exemplo, através do método *SMO* (Otimização mínima sequencial) proposto por *John Platt* [Platt (1998)].

2.3 CLASSIFICAÇÃO NÃO LINEAR CENTRALIZADA

Em diversas aplicações práticas os dados de treino $(x_i, y_i) \in S$ são distribuídos de forma não linear no espaço de origem $\mathbb{R}^n$, e nesse caso pode não haver um hiperplano que separe os dados de forma satisfatória. Nesses casos, a ideia é criar margens não lineares para separar esses dados entre suas classes distintas.

Para que o SVM possa ser extendido à classificão não linear, procura-se criar um mapeamento $\phi : \mathbb{R}^n \mapsto \mathcal{F}$ em que $\mathcal{F}$ é um espaço de *Hilbert* separável, tal que os dados do conjunto

$$\bar{S} = \{(\phi(x), y) \mid (x, y) \in S\}$$

[4] Fato que se deve a condição de folgas complementares, ver seção 11.8 de Luenberger e Ye (2016).

podem ser apartados por um hiperplano em $\mathcal{F}$ de acordo com a classe y. Dessa forma, seria possível reescrever o problema de suporte vetorial no espaço $\mathcal{F}$. De fato, a partir de (2.20), tem-se que

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \mu_i - \sum_{i=1}^{m}\sum_{j=1}^{m} \mu_i \mu_j y_i y_j \left\langle \phi(x_i), \phi(x_j) \right\rangle_{\mathcal{F}} \\ \text{s.a.} \quad & 0 \leqslant \mu_i \leqslant C \qquad\qquad i = 1, \cdots, m \\ & \sum_{i=1}^{m} \mu_i y_i = 0. \end{aligned} \tag{2.25}$$

Entretanto, $\mathcal{F}$ pode apresentar dimensão arbitrariamente grande mesmo que este seja finitamente gerado. Fato que leva a um conjunto de problemas comumente conhecido como *Curse of Dimensionality*[5]. Dentre esses problemas há, por exemplo, o *overfitting* do modelo devido a esparsidade dos dados em altas dimensões, a ineficiência da distinção dos dados através da métrica Euclidiana [Aggarwal, Hinneburg e Keim (2001)] e, principalmente, o aumento da complexidade computacional, o qual deriva diretamente da manipulação das variáveis no espaço $\mathcal{F}$.

Portanto, procura-se evitar o cálculo explícito da imagem dos vetores $x_i \in S$ no espaço $\mathcal{F}$ através do mapeamento ϕ. De fato, isso é possível observando que em (2.25) há apenas a necessidade de conhecer o produto interno definido no espaço $\mathcal{F}$. Dessa forma, considera-se $\mathcal{F}$ um espaço de *Hilbert* reproduzível por núcleos (Definição 10), em que o mapeamento ϕ é definido a partir do núcleo associado a esse espaço, ou seja[6],

$$\phi(x) = K(\cdot, x).$$

Observe que, o Teorema 3 garante que qualquer função $K : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ simétrica e positiva define um núcleo associado a um único espaço de *Hilbert* reproduzido por esse núcleo. Ou seja, ao invés de conhecer a forma explícita de $\phi(x_i)$, define-se uma função $K : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ simétrica e positiva, e busca-se resolver

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \mu_i - \sum_{i=1}^{m}\sum_{j=1}^{m} \mu_i \mu_j y_i y_j K(x_i, x_j) \\ \text{s.a.} \quad & 0 \leqslant \mu_i \leqslant C \qquad\qquad i = 1, \cdots, m \\ & \sum_{i=1}^{m} \mu_i y_i = 0. \end{aligned} \tag{2.26}$$

É importante notar que a solução do *SVM* computado em $\mathcal{F}$ é expressa de forma finita,

5 Termo introduzido por *Bellman* em Bellman (1957).

6 Note que $\phi(x)$ é um funcional linear, até mesmo quando os dados de treino são satisfatoriamente separados em um espaço real de dimensão finita, a saber $\mathbb{R}^p$, nesse caso $\mathcal{F}$ é tomado como o espaço dual de $\mathbb{R}^p$.

apesar deste espaço poder ser infinitamente gerado. De fato,

$$w = \sum_{i=1}^{m} \mu_i y_i \phi(x_i), \tag{2.27}$$

em que $w \in \mathcal{F}$. Ademais, a função para classificação de um dado vetor x é dada em função do núcleo K, a saber

$$\operatorname{sign}\left(\sum_{i=1}^{m} \mu_i y_i K(x_i, x) + b\right). \tag{2.28}$$

Observe que não há a necessidade de conhecer o mapa ϕ para classificar um vetor x.

Apesar de haver inúmeras funções simétricas e positivas, somente algumas dessas são vastamente empregadas em situações práticas por representarem núcleos associados a espaços comumente utilizados, a saber

Tabela 1 – Núcleos de interesse prático.

Linear	$k(x_i, x_j) = x_i^T x_j$
Polinomial	$k(x_i, x_j) = (x_i^T x_j + c)^d \quad d \in \mathbb{N}$ e $c \geqslant 0$
RBF	$k(x_i, x_j) = \exp\left(-\gamma \lvert\lvert x_i - x_j \rvert\rvert^2\right) \quad \gamma > 0$
RBF Laplaciano	$k(x_i, x_j) = \exp\left(-\gamma \lvert\lvert x_i - x_j \rvert\rvert\right) \quad \gamma > 0$

2.4 MÁQUINA DE SUPORTE VETORIAL DISTRIBUÍDO

O foco agora é tratar de problemas de suporte vetorial distribuído. Neste caso, o conjunto de dados pertencente ao treino do *SVM* é intrinsecamente descentralizado e distribuído entre diversos agentes (ou nós) de um rede necessariamente conexa. Ou seja, essa rede é representável por um grafo em que sempre há um caminho que conecta um nó a outro. Ainda, será considerado que as conexões entre os nós são feitas de forma bidirecional. Por fim, é importante ressaltar que os algoritmos desenvolvidos aqui são baseados no artigo Forero, Cano e Giannakis (2010).

Dessa forma, busca-se determinar um método alternativo para solucionar o *SVM* que seja aplicável a esse problema e apresente um modelo preditivo tão acurado quanto o derivado da situação em que os dados estão todos centralizados (subseção 2.2.3). Para tanto o problema distribuído será formulado de forma a forçar o consenso entre os agentes da rede com respeito a solução do problema de suporte vetorial. Nesse contexto, serão analisados tanto a classificação linear quanto a não linear.

2.4.1 *Aplicação à Classificação Linear*

Inicialmente, será abordada a classificação linear sob o contexto distribuído. Com isso, o problema de suporte vetorial (2.13) definido para o caso centralizado será

reformulado, para isso serão introduzidas novas variáveis. A saber, define-se N como o número de nós de uma rede em que cada nó comporta m_i dados de treino de forma descentralizada. Ademais, considera-se que cada nó determina um hiperplano descrito por um vetor (w_i, b_i). Logo, o problema (2.13) pode ser reescrito como

$$\begin{aligned}
\min \quad & \frac{1}{2}\sum_{i=1}^{N}||w_i||^2 + C\sum_{i=1}^{N}\sum_{j=1}^{m_i}\varepsilon_{ij} & & & \\
\text{s.a.} \quad & 1 - y_{ij}(x_{ij}^T w_i + b_i) - \varepsilon_{ij} \leqslant 0 & i = 1,\cdots,N & j = 1,\cdots,m_i \\
& -\varepsilon_{ij} \leqslant 0 & i = 1,\cdots,N & j = 1,\cdots,m_i \\
& w_i - w_k = 0 & i = 1,\cdots,N & k \in \mathbb{K}_i \\
& b_i - b_k = 0 & i = 1,\cdots,N & k \in \mathbb{K}_i \\
& (w_i, b_i) \in \mathbb{R}^{n+1} & i = 1,\cdots,N, &
\end{aligned} \tag{2.29}$$

em que os índices i e j representam, respectivamente, o nó e o dado que se refere determinada variável, por exemplo, ε_{ij} representa o erro do dado j pertencente ao nó i. Ademais, $w_i - w_k = 0$ e $b_i - b_k = 0$ são as restrições consensuais que garantem a troca de mensagem entre os nós de forma a garantir que o mesmo hiperplano seja utilizado em todos os nós. Note que essas restrições são impostas somente entre os vizinhos de cada nó i. Pois, isso é o suficiente para que haja consenso dentre todos os nós da rede, uma vez que a mesma é conexa. Neste caso é introduzido o conjunto $\mathbb{K}_i$ que representa os índices dos nós vizinhos ao nó i.

Com efeito, como a rede é conexa, dado nós arbitrários p e l sempre existe um caminho que os conecta. Em particular, o conjunto $\mathbb{K}_i$ para $i = 1,\cdots,N$ é sempre não vazio. Portanto o consenso local em $\mathbb{K}_p$ de uma solução viável (w_p, b_p) é consenso entre toda a vizinhança que o caminho entre p e l abrange. Mas, como p e l são arbitrários, qualquer solução viável de (2.29) é consensual entre toda a rede, ou seja, $w_1 = \cdots = w_M = w$ e $b_1 = \cdots = b_M = b$. Assim, note que qualquer vetor viável de (2.29) satisfaz o seguinte problema

$$\begin{aligned}
\min \quad & \frac{N}{2}||w||^2 + C\sum_{i=1}^{N}\sum_{j=1}^{m_i}\varepsilon_{ij} & & & \\
\text{s.a.} \quad & 1 - y_{ij}(x_{ij}^T w + b) - \varepsilon_{ij} \leqslant 0 & i = 1,\cdots,N & j = 1,\cdots,m_i \\
& -\varepsilon_{ij} \leqslant 0 & i = 1,\cdots,N & j = 1,\cdots,m_i \\
& (w, b) \in \mathbb{R}^{n+1}, & &
\end{aligned} \tag{2.30}$$

o qual é justamente um problema de suporte vetorial centralizado a menos de uma constante N, como descrito em (2.13).

2.4.1.1 Formulação do Algoritmo Distribuído

Definido o problema, procura-se criar um algoritmo que seja facilmente distribuído entre os agentes da rede e não necessite da comunicação dos dados de

treino. Para isso, busca-se aplicar o método *ADMM* ao problema (2.29), o que exige reescrevê-lo de forma a possuir a estrutura apresentada em (1.37). A priori, será introduzida uma nova notação, a saber

$$\begin{array}{lll} \min & \frac{1}{2}\sum_{i=1}^{N} v_i^T(I - e_{n+1}e_{n+1}^T)v_i + C\sum_{i=1}^{N}\mathcal{E}_i^T 1_i & \\ \text{s.a.} & 1_i - Y_i X_i v_i - \mathcal{E}_i \leqslant 0 & i = 1,\cdots,N \\ & -\mathcal{E}_i \leqslant 0 & i = 1,\cdots,N \\ & v_i - u_{ik} = 0 & i = 1,\cdots,N \;\; k \in \mathbb{K}_i \\ & v_k - u_{ik} = 0 & i = 1,\cdots,N \;\; k \in \mathbb{K}_i \\ & v_i \in \mathbb{R}^{n+1} & i = 1,\cdots,N, \end{array} \tag{2.31}$$

em que $v_i = [w_i^T \; b_i]^T$, e_{n+1} é o vetor canônico com coordenada $n+1$ unitária e $I \in \mathbb{R}^{n+1\,\mathrm{x}\,n+1}$ é a matriz identidade. Ademais,

$$\begin{array}{rcll} Y_i & = & \mathrm{diag}(y_{i1},\cdots,y_{im_i}) & \in \mathbb{R}^{m_i\,\mathrm{x}\,m_i} \\ X_i^T & = & \begin{bmatrix} x_{i1} & \cdots & x_{im_i} \\ 1 & \cdots & 1 \end{bmatrix} & \in \mathbb{R}^{n+1\,\mathrm{x}\,m_i} \\ \mathcal{E}_i & = & [\varepsilon_{i1} \; \cdots \; \varepsilon_{im_i}]^T & \in \mathbb{R}^{m_i} \\ 1_i & = & [1 \; \cdots \; 1]^T & \in \mathbb{R}^{m_i}. \end{array}$$

Note que a redundância introduzida pela variável u_{ik} é utilizada para desacoplar a variável v_i das variáveis v_k determinadas pelos vizinhos do nó i. Em particular, o problema (2.31) pode ser abordado a partir do método *ADMM*.

De fato, das restrições $v_i = u_{ik}$ para todos $i = 1,\cdots,N$ e $k \in \mathbb{K}_i$, tem-se que

$$\begin{array}{c} \{v_1 = u_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N = u_{Nk}\}_{k\in\mathbb{K}_N}. \end{array} \tag{2.32}$$

Logo correspondem a $\sum_{i=1}^{N} \#\mathbb{K}_i$ equações vetoriais. Como a rede é conexa, a soma das cardinalidades dos conjuntos $\mathbb{K}_i$ para todo nó i é exatamente o dobro do número de arestas do grafo que traduz a rede em questão[7], ou seja

$$\sum_{i=1}^{N} \#\mathbb{K}_i = 2E, \tag{2.33}$$

em que E representa o número de arestas do grafo. Como cada vetor v está em $\mathbb{R}^{n+1}$, então há $2E(n+1)$ restrições de igualdade. Dessa forma, (2.32) pode ser reescrita da seguinte forma

$$\hat{A}v = u, \tag{2.34}$$

[7] Teorema 1.1 de Bondy e Murty (2008).

em que

$$\hat{A} \;=\; \text{diag}(A_1, \cdots, A_N) \;\in \mathbb{R}^{2E(n+1)\,\times\, N(n+1)} \tag{2.35}$$

é uma matriz diagonal por blocos com $A_i = [I \;\cdots\; I]^T \in \mathbb{R}^{\#\mathbb{K}_i(n+1)\,\times\, n+1}$ e

$$v \;=\; \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \qquad \in \mathbb{R}^{N(n+1)}$$

$$u \;=\; \begin{bmatrix} \{u_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{u_{Nk}\}_{k\in\mathbb{K}_N} \end{bmatrix} \in \mathbb{R}^{2E(n+1)}.$$

Por outro lado, tomando das restrição $v_k = u_{ik}$ para todos $i = 1, \cdots, N$ e $k \in \mathbb{K}_i$, tem-se que

$$\begin{gathered} \{v_k = u_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_k = u_{Nk}\}_{k\in\mathbb{K}_N}. \end{gathered} \tag{2.36}$$

Note que é possível reordenar estas equações como segue

$$\begin{gathered} \{v_1 = u_{k1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N = u_{kN}\}_{k\in\mathbb{K}_N}. \end{gathered} \tag{2.37}$$

De fato, seja $a \in \mathbb{K}_i$ para um nó i qualquer, então

$$v_a = u_{ia} \;\in\; \{v_k = u_{ik}\}_{k\in\mathbb{K}_i},$$

em que o conjunto $\{v_k = u_{ik}\}_{k\in\mathbb{K}_i}$ necessariamente integra (2.36). Observando que a rede é conexa e bidirecional, então $i \in \mathbb{K}_a$. Logo, de forma análoga,

$$v_i = u_{ai} \;\in\; \{v_k = u_{ak}\}_{k\in\mathbb{K}_a},$$

em que o conjunto $\{v_i = u_{ik}\}_{k\in\mathbb{K}_i}$ também integra (2.36). Como o índice a é arbitrário, então

$$\begin{aligned} v_i = u_{ai} \;&\in\; \{v_k = u_{ak}\}_{k\in\mathbb{K}_a} \quad \forall a \in \mathbb{K}_i \\ &\Rightarrow\; \{v_i = u_{ai}\}_{a\in\mathbb{K}_i}. \end{aligned}$$

Portanto, é possível extrair de (2.36) o conjunto $\{v_i = u_{ki}\}_{k\in\mathbb{K}_k}$ para qualquer nó i da rede. Logo, o conjunto de equações (2.36) pode ser reordenado como apresentado em (2.37).

Dessa forma, (2.37) pode ser reescrita como

$$\hat{A}v = \hat{u}, \tag{2.38}$$

em que

$$\hat{u} = \begin{bmatrix} \{u_{k1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{u_{kN}\}_{k\in\mathbb{K}_N} \end{bmatrix} \in \mathbb{R}^{2E(n+1)}.$$

Por outro lado, observe que u pode ser transformado em $\hat{u}$ a partir da permutação entre os vetores u_{ik} e u_{ki}. Dessa forma é possível determinar uma matriz permutação $P \in \mathbb{R}^{2E(n+1)\times 2E(n+1)}$ tal que

$$\hat{u} = Pu. \tag{2.39}$$

A saber, $P = T \otimes I$[8] com $I \in \mathbb{R}^{n+1}$ e

$$T = \left[\{t_{1k}\}_{k\in\mathbb{K}_1}, \cdots, \{t_{Nk}\}_{k\in\mathbb{K}_N}\right] \quad \in \mathbb{R}^{2E\times 2E}, \tag{2.40}$$

em que[9]

$$t_{ik} = \begin{bmatrix} \{\delta_{i-i',\,k-1}\}_{i'\in\mathbb{K}_1} \\ \vdots \\ \{\delta_{i-i',\,k-N}\}_{i'\in\mathbb{K}_N} \end{bmatrix} \quad \in \mathbb{R}^{2E}. \tag{2.41}$$

Note que, de fato a matriz T representa uma permutação em que cada coluna j indica em que posição o j-ésimo vetor que integra u será realocado para formar $\hat{u}$. Com efeito, os deltas de *Kronecker* são responsáveis por garantir a permutação desejada entre u_{ik} e u_{ki}. Ademais, o produto de *Kronecker* utilizado na definição da matriz P ajusta a dimensionalidade da matriz T, uma vez que os objetos de interesse à permutação são vetores de dimensão $n+1$.

Portanto, as restrições

$$\begin{aligned} v_k &= u_{ik} \quad i = 1, \cdots, N \quad k \in \mathbb{K}_i \\ v_i &= u_{ik} \quad i = 1, \cdots, N \quad k \in \mathbb{K}_i \end{aligned}$$

podem ser apresentada da seguinte forma condensada

$$Av - Bu = 0, \tag{2.42}$$

8 O simbolo $\otimes$ representa o produto de *Kronecker*, o qual é dado por

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \vdots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{pm\times qn},$$

em que $A \in \mathbb{R}^{m\times n}$ e $B \in \mathbb{R}^{p\times q}$.

9 Já $\delta_{i,j}$ representa o delta de *Kronecker*, o qual é dado por

$$\delta_{i,j} = \begin{cases} 0 & \text{se } i \neq j \\ 1 & \text{se } i = j \end{cases}.$$

em que

$$A = \begin{bmatrix} \hat{A} \\ \hat{A} \end{bmatrix} \in \mathbb{R}^{4E(n+1) \times N(n+1)}$$

$$B = \begin{bmatrix} I \\ P \end{bmatrix} \in \mathbb{R}^{4E(n+1) \times 2E(n+1)}$$

com $I \in \mathbb{R}^{2E(n+1) \times 2E(n+1)}$.

Por outro lado, note que as restrições de desigualdade $1_i - Y_i X_i v_i \leqslant \mathcal{E}_i$ e $\mathcal{E}_i \geqslant 0$ para $i = 1, \cdots, N$ junto a necessidade de minimizar $\sum_{i=1}^{N} 1_i \mathcal{E}_i$, implicam que

$$\mathcal{E}_i = \max\{0, 1_i - Y_i X_i v_i\}. \tag{2.43}$$

De fato, como o objetivo é minimizar a soma dos erros e estes são limitados inferiormente, então $\mathcal{E}_i$ seria o próprio limite inferior, a saber $1_i - Y_i X_i v_i$. Entretanto, $\mathcal{E}_i$ ainda precisa ser não negativo, o que implica na igualdade (2.43).

Dessa forma, a partir de (2.43) e (2.42), o problema (2.31) pode ser reescrito como

$$\begin{aligned} \min \quad & F(v) \\ \text{s.a.} \quad & Av - Bu = 0 \\ & v \in \mathbb{R}^{N(n+1)} \\ & u \in \mathbb{R}^{2E(n+1)}, \end{aligned} \tag{2.44}$$

em que

$$F(v) = \sum_{i=1}^{N} f_i(v_i),$$

com

$$f_i(v_i) = \frac{1}{2} v_i^T (I - e_{n+1} e_{n+1}^T) v_i + C 1_i^T \max\{0, 1_i - Y_i X_i v_i\}, \tag{2.45}$$

que é claramente convexa. Portanto, de fato é possível enquadrar o problema de suporte vetorial distribuído (2.31) a um caso particular da classe de problemas de interesse ao método *ADMM* (1.37).

Assim, pode-se solucionar o problema (2.44) através do método *ADMM* e, a partir da Proposição 8, é garantido que todo ponto de acumulação da sequência gerada por esse método é solução do problema. A saber, aplicando o *ADMM* a (2.44), tem-se que

$$\begin{aligned} v^{l+1} &\in \arg\min_{v \in \mathbb{R}^{N(n+1)}} \mathcal{L}_c(v, u^l, \lambda^l) \\ u^{l+1} &\in \arg\min_{u \in \mathbb{R}^{2E(n+1)}} \mathcal{L}_c(v^{l+1}, u, \lambda^l) \\ \lambda^{l+1} &= \lambda^l + c(Av^{l+1} - Bu^{l+1}), \end{aligned} \tag{2.46}$$

em que $\mathcal{L}_c$ é o lagrangeano aumentado do problema (2.44), ou seja,

$$\mathcal{L}_c(v, u, \lambda) = F(v) + (Av - Bu)^T \lambda + \frac{c}{2} ||Av - Bu||^2. \tag{2.47}$$

Entretanto, note que o método apresentado em (2.46) não é claramente paralelizável, pois não parece ser separado em tarefas independentes. Dessa forma, busca-se resgatar os somatórios que evidenciam a característica distribuída e descentralizada do problema original. Para tanto, serão introduzidas novas notações para diferentes blocos que integram a variável dual de forma a corresponder as $4E$ restrições vetoriais. A saber,

$$\lambda = \begin{bmatrix} \{\lambda_{1k_1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\lambda_{Nk_1}\}_{k\in\mathbb{K}_N} \\ \{\lambda_{k1_2}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\lambda_{kN_2}\}_{k\in\mathbb{K}_N} \end{bmatrix}. \tag{2.48}$$

Assim, observe que o produto interno e o termo de regularização provenientes do lagrangeano aumentado $\bar{\mathcal{L}}_c$ podem ser reescritos como

$$\begin{aligned} (Av-Bu)^T\lambda &= \begin{bmatrix} \{v_1-u_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N-u_{Nk}\}_{k\in\mathbb{K}_N} \\ \{v_1-u_{k1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N-u_{kN}\}_{k\in\mathbb{K}_N} \end{bmatrix}^T \begin{bmatrix} \{\lambda_{1k_1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\lambda_{Nk_1}\}_{k\in\mathbb{K}_N} \\ \{\lambda_{k1_2}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\lambda_{kN_2}\}_{k\in\mathbb{K}_N} \end{bmatrix} \\ ||Av-Bu||^2 &= \begin{bmatrix} \{v_1-u_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N-u_{Nk}\}_{k\in\mathbb{K}_N} \\ \{v_1-u_{k1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N-u_{kN}\}_{k\in\mathbb{K}_N} \end{bmatrix}^T \begin{bmatrix} \{v_1-u_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N-u_{Nk}\}_{k\in\mathbb{K}_N} \\ \{v_1-u_{k1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{v_N-u_{kN}\}_{k\in\mathbb{K}_N} \end{bmatrix}, \end{aligned} \tag{2.49}$$

em que apenas foram utilizadas a identidade proveniente de $Av-Bu$ e as novas variáveis duais introduzidas em (2.48). Desa forma, tem-se que

$$\begin{aligned} (Av-Bu)^T\lambda &= \sum_{i=1}^{M}\sum_{k\in\mathbb{K}} \lambda_{ik_1}^T(v_i-u_{ik}) + \lambda_{ki_2}^T(v_i-u_{ki}) \\ ||Av-Bu||^2 &= \sum_{i=1}^{M}\sum_{k\in\mathbb{K}} \frac{c}{2}||v_i-u_{ik}||^2 + \frac{c}{2}||v_i-u_{ki}||^2. \end{aligned} \tag{2.50}$$

Ademais, a partir da identidade (2.43), será reintroduzida a variável $\mathcal{E}_i$. Dessa forma, tem-se que

$$F(v,\mathcal{E}) = \sum_{i=1}^{N} f_i(v_i,\mathcal{E}_i), \tag{2.51}$$

em que $f_i(v_i,\mathcal{E}_i) = \frac{1}{2}v_i^T(I - e_{n+1}e_{n+1}^T)v_i + C1_i^T\mathcal{E}_i$ e

$$\mathcal{E} = \begin{bmatrix} \mathcal{E}_1 \\ \vdots \\ \mathcal{E}_N \end{bmatrix} \in \mathbb{R}^{Nm}$$

com $m = \sum_{i=1}^{N} m_i$.

Portanto, o lagrangeano aumentado $\mathcal{L}_c$, pode ser reescrito como

$$\begin{aligned} \mathcal{L}_c\Big((v,\mathcal{E}),u,\lambda\Big) = \sum_{i=1}^{N}\Bigg\{ & f(v_i,\mathcal{E}_i) + \sum_{k\in\mathbb{K}_i}\Big[(v_i-u_{ik})^T\lambda_{ik_1} \\ & +(v_i-u_{ki})^T\lambda_{ki_2} \quad + \frac{c}{2}||v_i-u_{ik}||^2 + \frac{c}{2}||v_i-u_{ki}||^2\Big]\Bigg\}, \end{aligned} \tag{2.52}$$

em que u_{ki} e λ_{ki_2} são, respectivamente, interpretadas como as variáveis auxiliar e dual determinadas pelos nós vizinhos ao nó i.

Logo, o método apresentado em (2.46) é reescrito como

$$\begin{aligned} (v,\mathcal{E})^{l+1} &\in \arg\min_{(v,\mathcal{E})\in\mathbb{H}} \mathcal{L}_c\Big((v,\mathcal{E}),u^l,\lambda^l\Big) \\ u^{l+1} &\in \arg\min_{u\in\mathbb{R}^{2E(n+1)}} \mathcal{L}_c\Big((v,\mathcal{E})^{l+1},u,\lambda^l\Big) \\ \lambda^{l+1} &= \lambda^l + c(Av^{l+1} - Bu^{l+1}), \end{aligned} \tag{2.53}$$

em que

$$\mathbb{H} = \{(v,\mathcal{E}) \mid (v_i,\mathcal{E}_i) \in \mathbb{H}_i \quad i = 1,\cdots,N\}$$

com $\mathbb{H}_i = \{(v_i,\mathcal{E}_i) \mid 1_i - Y_iX_iv_i - \mathcal{E}_i \leqslant 0 \text{ e } -\mathcal{E}_i \leqslant 0\}$.

Dessa forma, observe que a minimização com respeito a variável $(v,\mathcal{E})$ é claramente separável entre as variáveis $\{(v_i,\mathcal{E}_i)\}_{i=1,\cdots,N}$. A saber, realiza-se a cada nó i a seguinte minimização

$$(v_i,\mathcal{E}_i)^{l+1} \in \arg\min_{(v_i,\mathcal{E}_i)\in\mathbb{H}_i} \mathcal{V}_c\Big((v_i,\mathcal{E}_i),\{u_{ik}^l,u_{ki}^l,\lambda_{ik_1}^l,\lambda_{ki_2}^l\}_{k\in\mathbb{K}_i}\Big) \tag{2.54}$$

com

$$\begin{aligned} &\mathcal{V}_c\Big((v_i,\mathcal{E}_i),\{u_{ik}^l,u_{ki}^l,\lambda_{ik_1}^l,\lambda_{ki_2}^l\}_{k\in\mathbb{K}_i}\Big) = \\ &f(v_i,\mathcal{E}_i) + \sum_{k\in\mathbb{K}_i}\Big[v_i^T\lambda_{ik_1}^l + v_i^T\lambda_{ki_2}^l + \frac{c}{2}||v_i-u_{ik}^l||^2 + \frac{c}{2}||v_i-u_{ki}^l||^2\Big]. \end{aligned} \tag{2.55}$$

Observe que a função $\mathcal{V}_c$ é definida como a parte pertinente de $\mathcal{L}_c$ à minimização com relação a variável $(v_i,\mathcal{E}_i)$. É importante ressaltar a exigência do conhecimento das variáveis $\{u_{ki}^l\}_{k\in\mathbb{K}_i}$ e $\{\lambda_{ki_2}^l\}_{k\in\mathbb{K}_i}$ por cada nó i, ou seja, a necessidade da troca de

mensagens entre qualquer nó i e seus nós vizinhos com respeito aos vetores $\{u_{ki}^l\}_{k\in\mathbb{K}_i}$ e $\{\lambda_{ki_2}^l\}_{k\in\mathbb{K}_i}$.

Ademais, a minimização sobre a variável u pode, também, ser feita separadamente para cada nó vizinho de um determinado nó i. Pois, a soma $\sum_{k\in\mathbb{K}_i}$ de (2.52) é separável entre as variáveis $\{u_{ik}\}_{k\in\mathbb{K}_i}$[10]. De fato, basta notar a seguinte reordenação nos somatórios

$$\begin{aligned}\sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}(v_i-u_{ki})^T\lambda_{ki_1} &= \sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}(v_k-u_{ik})^T\lambda_{ik_1}\\ \sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}\frac{c}{2}||v_i-u_{ki}||^2 &= \sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}\frac{c}{2}||v_k-u_{ik}||^2,\end{aligned} \tag{2.56}$$

em que é utilizado o mesmo raciocínio que permite ir de (2.37) para (1.38). Dessa forma, aplicando essas identidades a $\mathcal{L}_c$, a minimização segundo a variável u passa a ser trivialmente separável entre os nós da rede. A saber, cada nó i realiza o seguinte conjunto de minimizações

$$u_{ik}^{l+1} \in \arg\min_{u_{ik}\in\mathbb{R}^{n+1}} \mathcal{U}_c\left(v_i^{l+1}, v_k^{l+1}, u_{ik}, \lambda_{ik_1}^l, \lambda_{ik_2}^l\right) \quad \forall k\in\mathbb{K}_i \tag{2.57}$$

com

$$\begin{aligned}&\mathcal{U}_c\left(v_i^{l+1}, v_k^{l+1}, u_{ik}, \lambda_{ik_1}^l, \lambda_{ik_2}^l\right) =\\ &-(\lambda_{ik_1}^l)^T u_{ik} - (\lambda_{ik_2}^l)^T u_{ik} + \frac{c}{2}||v_i^{l+1}-u_{ik}||^2 + \frac{c}{2}||v_k^{l+1}-u_{ik}||^2.\end{aligned} \tag{2.58}$$

De forma análoga, $\mathcal{U}_c$ é definido como a parte de $\mathcal{L}_c$ pertinente à minimização com respeito a variável u_{ik} Por fim, note que é necessária a troca de mensagens entre cada nó i e seus vizinhos com respeito aos vetores $\{v_k^{l+1}\}_{k\in\mathbb{K}_i}$.

Portanto, a partir do método (2.53), é possível derivar o Algoritmo 6, em que a atualização da variável dual λ foi reescrita utilizando as notações introduzidas em (2.48).

Entretanto, esse algoritmo exige demasiado processamento devido ao grande número de variáveis auxiliares necessárias ao processo, demasiado *overhead* de comunicação devido ao grande tráfego de informações provenientes da comunicação dos vetores $\{u_{ki}^l\}_{k\in\mathbb{K}_i}$ e $\{\lambda_{ki_2}^l\}_{k\in\mathbb{K}_i}$ e, ainda, possui pouca praticidade na implementação. Dessa forma, procura-se simplificar o Algoritmo 6 para reduzir o número de variáveis a serem computadas e transmitidas.

Assim, observe que a minimização para determinar o vetor u_{ik}^{l+1} envolve um problema quadrático e irrestrito, como apresentado em (2.58). Logo é possível determinar uma forma fechada para o vetor u_{ik}^l. De fato, aplicando a condição de

[10] Observe que isso só é possível uma vez que, em (2.31), foram introduzidas as variáveis $\{u_{ik}\}_{i=1,\cdots,N \text{ e } k\in\mathbb{K}_i}$ para desacoplar os vetores $\{v_i\}_{i=1,\cdots,N}$ dos vetores $\{v_k\}_{k\in\mathbb{K}_i}$.

Algoritmo 6 – SVM Linear Distribuído (LDSVM): Versão 1

Dados: $\lambda_{ik_1}^0$, $\lambda_{ik_2}^0$ e u_{ik} para todo $i = 1, \cdots, N$ e $k \in \mathbb{K}_i$
para cada *nó* i **faça**
  para $l = 0, 1, 2, 3, \ldots$ **faça**

$$
\begin{array}{rcll}
(v_i, \mathcal{E}_i)^{l+1} & \in & \arg \min\limits_{(v_i,\mathcal{E}_i)\in\mathbb{H}_i} \mathcal{V}_c\Big((v_i,\mathcal{E}_i), \{u_{ik}^l, u_{ki}^l, \lambda_{ik_1}^l, \lambda_{ki_2}^l\}_{k\in\mathbb{K}_i}\Big) & \\
u_{ik}^{l+1} & \in & \arg \min\limits_{u_{ik}\in\mathbb{R}^{n+1}} \mathcal{U}_c\Big(v_i^{l+1}, v_k^{l+1}, u_{ik}, \lambda_{ik_1}^l, \lambda_{ik_2}^l\Big) & \forall k \in \mathbb{K}_i \\
\lambda_{ik_1}^{l+1} & = & \lambda_{ik_1}^l + c(v_i^{l+1} - u_{ik}^{l+1}) & \forall k \in \mathbb{K}_i \\
\lambda_{ki_2}^{l+1} & = & \lambda_{ki_2}^l + c(v_i^{l+1} - u_{ki}^{l+1}) & \forall k \in \mathbb{K}_i
\end{array}
$$

  fim
fim

primeira ordem, tem-se que

$$
\begin{aligned}
0 &= -(\lambda_{ik_1}^l + \lambda_{ik_2}^l) - c(v_i^{l+1} - u_{ik}^{l+1}) - c(v_k^{l+1} - u_{ik}^{l+1}) \\
u_{ik}^{l+1} &= \frac{1}{2c}(\lambda_{ik_1}^l + \lambda_{ik_2}^l) + \frac{1}{2}(v_i^{l+1} + v_k^{l+1}).
\end{aligned} \tag{2.59}
$$

Dessa expressão segue diretamente que

$$
u_{ki}^{l+1} = \frac{1}{2c}(\lambda_{ki_1}^l + \lambda_{ki_2}^l) + \frac{1}{2}(v_k^{l+1} + v_i^{l+1}). \tag{2.60}
$$

Logo, substituindo (2.59) e (2.60) nas expressões para as variáveis duais que aparecem no Algoritmo 6, deriva-se

$$
\lambda_{ik_1}^{l+1} = \frac{1}{2}(\lambda_{ik_1}^l - \lambda_{ik_2}^l) + \frac{c}{2}(v_i^{l+1} - v_k^{l+1}) \tag{2.61}
$$

$$
\lambda_{ki_2}^{l+1} = \frac{1}{2}(\lambda_{ki_2}^l - \lambda_{ki_1}^l) + \frac{c}{2}(v_i^{l+1} - v_k^{l+1}). \tag{2.62}
$$

Ainda, de (2.62) segue que

$$
\lambda_{ik_2}^{l+1} = \frac{1}{2}(\lambda_{ik_2}^l - \lambda_{ik_1}^l) + \frac{c}{2}(v_k^{l+1} - v_i^{l+1}). \tag{2.63}
$$

Observe que as expressões (2.61) e (2.63) garantem que $\lambda_{ik_1}^l = -\lambda_{ik_2}^l$ para toda iteração l com exceção do caso $l = 0$. Assim, procura-se inicializar as variáveis duais de forma que $\lambda_{ik_1}^0 = -\lambda_{ik_2}^0$, ou seja,

$$
\lambda_{ik_1}^0 = \lambda_{ik_2}^0 = 0 \qquad i = 1, \cdots, N \text{ e } \forall k \in \mathbb{K}_i. \tag{2.64}
$$

Dessa forma, é possível introduzir uma nova variável λ_{ik} tal que

$$
\lambda_{ik}^l = \lambda_{ik_1}^l = -\lambda_{ik_2}^l \qquad \forall l. \tag{2.65}
$$

Nesse caso, as expressões (2.59) e (2.61) podem ser reescritas como

$$u_{ik}^{l+1} = \frac{1}{2}(v_i^{l+1} + v_k^{l+1}) \tag{2.66}$$

$$\lambda_{ik}^{l+1} = \lambda_{ik}^{l} + \frac{c}{2}(v_i^{l+1} - v_k^{l+1}). \tag{2.67}$$

Com isso, note que, a partir da expressão (2.66), conclui-se que

$$u_{ik}^{l} = u_{ki}^{l} \quad \forall l. \tag{2.68}$$

Por outro lado, ainda, é possível derivar uma forma alternativa para λ_{ik}^{l+1} além de (2.67). De fato, somando a expressão (2.67) recursivamente sobre as iterações l, obtém-se que

$$\lambda_{ik}^{l+1} = \sum_{l=0}^{l+1} \frac{c}{2}(v_i^{l+1} - v_k^{l+1}). \tag{2.69}$$

Por fim, desse resultado é possível afirmar que

$$\lambda_{ik}^{l} = -\lambda_{ki}^{l} \quad \forall l. \tag{2.70}$$

Assim, utilizando a expressão para o vetor u_{ik} em (2.66) e as relações derivadas em (2.68), (2.65) e (2.70), é possível reescrever a função $\mathcal{V}_c$ referente à minimização de $(v_i, \mathcal{E}_i)$ (2.55) da seguinte forma

$$\mathcal{V}_c\Big((v_i,\mathcal{E}_i), v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l\Big) = f(v_i,\mathcal{E}_i) + 2v_i^T\lambda_i^l + c\sum_{k\in\mathbb{K}_i}\left\|v_i - \frac{1}{2}(v_i^l + v_k^l)\right\|^2, \tag{2.71}$$

em que $\lambda_i^l = \sum_{k\in\mathbb{K}_i} \lambda_{ik}^l$.

Com isso é possível simplificar o Algoritmo 6, obtendo o Algoritmo 7. Note que, dessa forma, evita-se a demanda excessiva de comunicação na rede,

Algoritmo 7 – SVM Linear Distribuído (LDSVM): Versão 2

Dados: $\lambda_{ik}^0 = 0$ e v_i^0 para todo $i = 1, \cdots, N$ e $k \in \mathbb{K}_i$
para cada *nó* i **faça**
  para $l = 0, 1, 2, 3, \ldots$ **faça**

$$(v_i,\mathcal{E}_i)^{l+1} \in \arg\min_{(v_i,\mathcal{E}_i)\in\mathbb{H}_i} \mathcal{V}_c\Big((v_i,\mathcal{E}_i), v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l\Big)$$

$$\lambda_i^{l+1} = \lambda_i^l + \frac{c}{2}\sum_{k\in\mathbb{K}_i}(v_i^{l+1} - v_k^{l+1})$$

  fim
fim

uma vez que não há mais a necessidade de comunicar os vetores $\{u_{ki}^l\}_{k\in\mathbb{K}_i}$ e $\{\lambda_{ki_2}^l\}_{k\in\mathbb{K}_i}$ entre um nó i qualquer e seus vizinhos. Note que nesse algoritmo a atualização da variável dual λ_i segue diretamente da expressão (2.67).

Apesar de não haver mais tráfego excessivo de dados na rede e a implementação ter sido simplificada, busca-se, ainda, tornar o problema mais prático de forma a desassociar o vetor v_i do $\mathcal{E}_i$ na minimização com respeito ao vetor $(v_i, \mathcal{E}_i)$. Para tanto, essa minimização será abordada a partir das condições de *Karush-Kuhn-Tucker (KKT)*[11], as quais são suficientes para determinar o minimizador desse problema. Pois, ele é um problema quadrático. Assim, o lagrangeano proveniente dessa minimização é dado por

$$\begin{aligned}&\mathcal{L}\Big((v_i,\,\mathcal{E}_i), v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l, \mu_i, \eta_i\Big) = \\ &f(v_i,\mathcal{E}_i) + 2v_i^T(\lambda_i^l) + c\sum_{k\in\mathbb{K}_i}\left\|v_i - \frac{1}{2}(v_i^l + v_k^l)\right\|^2 + \mu_i^T(1_i - Y_iX_iv_i - \mathcal{E}_i) - \eta_i^T\mathcal{E}_i,\end{aligned} \tag{2.72}$$

em que μ_i e η_i são as variáveis duais relacionadas às restrições impostas pelo conjunto $\mathbb{H}_i$. Note que, como as restrições são lineares, não há a necessidade de exigir regularidade[12] dos vetores a serem analisados através das condições *KKT*. A saber, o vetor $(v_i,\mathcal{E}_i)^{l+1}$ satisfaz $\inf\limits_{(v_i,\mathcal{E}_i)\in\mathbb{H}_i} \mathcal{V}_c\Big((v_i,\mathcal{E}_i), v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l\Big)$ se somente se $(v_i,\mathcal{E}_i)^{l+1}$ e $(\mu_i,\eta_i)^{l+1}$ satisfazem o seguinte sistema

$$\begin{cases} \nabla_{v_i}\mathcal{L}\Big((v_i,\,\mathcal{E}_i)^{l+1}, v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l, \mu_i^{l+1}, \eta_i^{l+1}\Big) &= 0 \\ \nabla_{\mathcal{E}_i}\mathcal{L}\Big((v_i,\,\mathcal{E}_i)^{l+1}, v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l, \mu_i^{l+1}, \eta_i^{l+1}\Big) &= 0 \\ 1_i - Y_iX_iv_i^{l+1} - \mathcal{E}_i^{l+1} &\leqslant 0 \\ -\mathcal{E}_i^{l+1} &\leqslant 0 \\ \mu_i^{l+1} &\geqslant 0 \\ \eta_i^{l+1} &\geqslant 0 \\ (1_i - Y_iX_iv_i^{l+1} - \mathcal{E}_i^{l+1})^T\mu_i^{l+1} &= 0 \\ (\eta_i^{l+1})^T\mathcal{E}_i^{l+1} &= 0 \end{cases}, \tag{2.73}$$

em que

$$\begin{aligned}&\nabla_{v_i}\mathcal{L}\Big((v_i,\,\mathcal{E}_i)^{l+1}, v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l, \mu_i^{l+1}, \eta_i^{l+1}\Big) = \\ &\quad (I - e_{n+1}e_{n+1}^T)v_i + 2\lambda_i^l + 2c\sum_{k\in\mathbb{K}_i}\left(v_i - \frac{1}{2}(v_i^l + v_k^l)\right) - X_i^TY_i\mu_i \\ &\nabla_{\mathcal{E}_i}\mathcal{L}\Big((v_i,\,\mathcal{E}_i)^{l+1}, v_i^l, \{v_k^l\}_{k\in\mathbb{K}_i}, \lambda_i^l, \mu_i^{l+1}, \eta_i^{l+1}\Big) = C1_i - \mu_i - \eta_i.\end{aligned} \tag{2.74}$$

A partir das expressões para os gradientes de $\mathcal{L}$ com relação a v_i e $\mathcal{E}_i$, é possível derivar, respectivamente, que

$$v_i^{l+1} = D_i^{-1}\left(X_i^TY_i\mu_i^{l+1} - r_i^l\right) \tag{2.75}$$

$$\eta_i^{l+1} = C1_i - \mu_i^{l+1}, \tag{2.76}$$

[11] Condições necessárias de primeira ordem para problemas de otimização com restrições. É apresentada na seção 11.8 de Luenberger e Ye (2016).

[12] Definição introduzida na seção 11.8 de Luenberger e Ye (2016), a qual impõe a independência linear entre os gradientes das restrições.

em que $r_i^l = 2\lambda_i^l - c\sum_{k\in\mathbb{K}_i}(v_i^l + v_k^l)$ e $D_i = (I - e_{n+1}e_{n+1}^T) + 2c\#\mathbb{K}_i I$. Ademais, é importante ressaltar que a matriz D_i é diagonal e sempre possui inversa, pois qualquer nó i possui ao menos um vizinho (a rede é conexa). Dessa forma, a partir da expressão (2.76), é possível concluir que $0 \leqslant \mu_i^{l+1} \leqslant C1_i$. De fato, como μ_i e η_i são não negativos, então

$$\begin{aligned} & C1_i = \eta_i^{l+1} + \mu_i^{l+1} \geqslant \mu_i^{l+1} \\ \Rightarrow \quad & 0 \leqslant \mu_i^{l+1} \leqslant C1_i. \end{aligned}$$

Por outro lado, substituindo (2.76) em $(\eta_i^{l+1})^T\mathcal{E}_i^{l+1} = 0$, tem-se que

$$(\mu_i^{l+1})^T\mathcal{E}_i^{l+1} = C1_i^T\mathcal{E}_i^{l+1} \tag{2.77}$$

Dessa forma, substitui-se essa identidade em $(1_i - Y_iX_iv_i^{l+1} - \mathcal{E}_i^{l+1})^T\mu_i^{l+1} = 0$ obtendo que

$$(1_i - Y_iX_iv_i^{l+1})^T\mu_i^{l+1} = C1_i^T\mathcal{E}_i^{l+1}. \tag{2.78}$$

Expressão que pode ser reescrita utilizando (2.75), a saber

$$-(\mu_i^{l+1})^TY_iX_iD_i^{-1}X_i^TY_i\mu_i^{l+1} + (1_i + Y_iX_iD_i^{-1}r_i^l)^T\mu_i^{l+1} = C1_i^T\mathcal{E}_i^{l+1} \tag{2.79}$$

Agora, considere o conjunto de vetores $(v_i, \mathcal{E}_i, \mu_i, \eta_i)$ tais que satisfazem o sistema (2.73) com exceção das folgas complementares, dadas por

$$\begin{aligned} (1_i - Y_iX_iv_i - \mathcal{E}_i)^T\mu_i &= 0 \\ (1_i - Y_iX_iv_i - \mathcal{E}_i)^T\mu_i &= 0. \end{aligned}$$

Dessa forma, analogamente ao realizado anteriormente, tem-se que

$$v_i = D_i^{-1}\left(X_i^TY_i\mu_i - r_i^l\right) \tag{2.80}$$

$$0 \leqslant \mu_i^{l+1} \leqslant C1_i. \tag{2.81}$$

Utilizando a viabilidade de v_i e o fato de $\mu_i \geqslant 0$, sabe-se que

$$(1_i - Y_iX_iv_i)^T\mu_i \leqslant \mu_i^T\mathcal{E}_i.$$

Ademais, observando que μ_i satisfaz a expressão (2.81), tem-se que

$$(1_i - Y_iX_iv_i)^T\mu_i \leqslant C1_i^T\mathcal{E}_i,$$

em particular vale que

$$(1_i - Y_iX_iv_i)^T\mu_i \leqslant C1_i^T\mathcal{E}_i^{l+1}.$$

Substituindo v_i pela expressão (2.80), essa desigualdade é reescrita como

$$-\mu_i^TY_iX_iD_i^{-1}X_i^TY_i\mu_i + (1_i + Y_iX_iD_i^{-1}r_i^l)^T\mu_i \leqslant C1_i^T\mathcal{E}_i^{l+1}. \tag{2.82}$$

Por fim, note que (2.79) e (2.82) implicam que

$$\mu_i^{l+1} \in \arg\max_{0\leqslant\mu_i\leqslant C1_i} \left\{ -\frac{1}{2}\mu_i^T Y_i X_i D_i^{-1} X_i^T Y_i \mu_i + (1_i + Y_i X_i D_i^{-1} r_i^l)^T \mu_i \right\}. \tag{2.83}$$

Assim, a partir de (2.83) e (2.75), o Algoritmo 7 pode ser reescrito da seguinte forma

Algoritmo 8 – SVM Linear Distribuído (LDSVM)

Dados: $\lambda_i^0 = 0$ e v_i^0 para todo $i = 1, \cdots, N$
para cada *nó* i **faça**
  para $l = 0, 1, 2, 3, \ldots$ **faça**

$$\begin{aligned}
\mu_i^{l+1} &\in \arg\max_{0\leqslant\mu_i\leqslant C1_i} \left\{ -\frac{1}{2}\mu_i^T Y_i X_i D_i^{-1} X_i^T Y_i \mu_i + (1_i + Y_i X_i D_i^{-1} r_i^l)^T \mu_i \right\} \\
v_i^{l+1} &= D_i^{-1}\left(X_i^T Y_i \mu_i^{l+1} - r_i^l \right) \\
\lambda_i^{l+1} &= \lambda_i^l + \frac{c}{2}\sum_{k\in\mathbb{K}_i} (v_i^{l+1} - v_k^{l+1})
\end{aligned}$$

com

$$\begin{aligned}
r_i^l &= 2\lambda_i^l - c\sum_{k\in\mathbb{K}_i} (v_i^l + v_k^l) \\
D_i &= (I - e_{n+1}e_{n+1}^T) + 2c\#\mathbb{K}_i I
\end{aligned}$$

  fim
fim

Ademais, é importante ressaltar que o problema de maximização para determinar a variável dual μ_i (2.83) é um problema quadrático restrito a uma caixa, o qual pode ser abordado através da projeção de gradientes, por exemplo, utilizando o método *GENCAN* [Birgin e Martínez (2002)].

Em suma, note que os algoritmos Algoritmo 6, 7 e 8 nada mais são que manipulações do *ADMM* aplicados ao problema de suporte vetorial distribuído descentralizado (2.29). Portanto, pela Proposição 8, esses algoritmos convergem a solução do problema (2.29). A qual, como discutido no início dessa seção 2.4, é a mesma que a solução determinada para o problema centralizado apresentado em (2.13).

2.4.2 *Aplicação à Classificação Não Linear*

Até aqui foram abordados aspectos pertinentes à classificação linear distribuída com os dados de treino descentralizados. Para tanto, o problema centralizado de máquina de suporte vetorial foi reformulado como apresentado em (2.29). Entretanto,

essa formulação é impraticável para abordar o *SVM* distribuído aplicado à classificação não linear. De fato, como apresentado em seção 2.3, determina-se uma função $K : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ simétrica e positiva, a qual necessariamente é o núcleo associado a um espaço *RKHS* denotado por $\mathcal{F}^M$ (Teorema 3), em que $M \in [1, \infty]$ determina a dimensão do espaço reproduzido por núcleos[13]. Ademais, define-se o mapa $\phi : \mathbb{R}^n \mapsto \mathcal{F}^M$ como $\phi(x) = K(\cdot, x)$. Dessa forma, ao definir a função K e mapear as variáveis de interesse a formulação (2.29) sobre $\mathcal{F}^M$, torna-se impraticável ou impossível solucionar (2.29). Pois, as restrições consensuais impostas por essa formulação exigem a comunicação de $\omega_i \in \mathcal{F}^M$ para todo nó i, fato que implica na computação de vetores com dimensionalidade arbitrariamente grande ou, até mesmo, infinita.

Assim, busca-se outra formulação tal que evite a computação explícita de vetores no espaço $\mathcal{F}^M$. Para isso, será utilizada a abordagem proposta em Forero, Cano e Giannakis (2010), a qual ao invés de exigir o consenso entre as soluções determinadas pelos agentes da rede, é exigido apenas que os agentes concordem quanto a transformação linear dada por $\Phi_\chi \omega_i$, em que é definido

$$\Phi_\chi = \begin{bmatrix} \phi(\chi_1)^T \\ \vdots \\ \phi(\chi_p)^T \end{bmatrix} \in \mathcal{F}^{p \times M}$$

com $\{\chi_i\}_{i=1,\cdots,p}$ sendo vetores em $\mathbb{R}^n$ comuns a todo nó. Note que, dessa forma, diferente do caso linear, os agentes não necessariamente determinam o mesmo classificador, mas é garantido que os diferentes modelos preditivos determinados a cada nó concordem com a classificação de um conjunto finito de p vetores. Além disso, p é tomado de forma que seja necessariamente menor que M. Pois, busca-se evitar a manipulação de vetores em espaços de alta dimensionalidade.

A saber, o problema de suporte vetorial distribuído pode ser reformulado para o caso não linear da seguinte forma

$$\begin{aligned}
\min \quad & \frac{1}{2}\sum_{i=1}^{N} ||\omega_i||^2_{\mathcal{F}^M} + C\sum_{i=1}^{N} \mathcal{E}_i^T 1_i & & & \\
\text{s.a.} \quad & 1 - Y_i(\Phi_{X_i}\omega_i + b_i 1_i) - \mathcal{E}_i \leqslant 0 & i = 1, \cdots, N & & \\
& -\mathcal{E}_i \leqslant 0 & i = 1, \cdots, N & & \\
& \Phi_\chi \omega_i - \Phi_\chi \omega_k = 0 & i = 1, \cdots, N & k \in \mathbb{K}_i & \\
& b_i - b_k = 0 & i = 1, \cdots, N & k \in \mathbb{K}_i & \\
& (\omega_i, b_i) \in \mathcal{F}^M \times \mathbb{R} & i = 1, \cdots, N, & &
\end{aligned} \tag{2.84}$$

[13] Note que a noção de dimensão para espaços reproduzidos por núcleos só existe pois os mesmos são separáveis.

em que

$$\Phi_{X_i} = \begin{bmatrix} \phi(x_{i1})^T \\ \vdots \\ \phi(x_{im_i})^T \end{bmatrix} \in \mathcal{F}^{m_i \times M},$$

de forma que $\Phi_\chi \omega_i \in \mathbb{R}^p$ e $\Phi_{X_i} \omega_i \in \mathbb{R}^{m_i}$, com $\langle \phi(x), \omega_i \rangle_{\mathcal{F}^M} = \phi(x)^T \omega_i$. É importante ressaltar que o mapa ϕ é definido a partir do núcleo K associado a $\mathcal{F}^M$, a saber $\phi(x) = K(\cdot, x)$. Ainda, note que o problema (2.84) define, para cada nó, um hiperplano em $\mathcal{F}^M$ a partir do vetor ω_i e do escalar b_i. Dessa forma, é possível classificar localmente qualquer vetor $x \in \mathbb{R}^n$ a partir da função

$$g_i(x) = f_i(x) + b_i, \tag{2.85}$$

em que é definido $f_i(x) = \langle \phi(x), \omega_i \rangle_{\mathcal{F}^M}$ apenas para simplificar a notação. Observe que a classe y de x é determinada apenas pelo sinal de $g_i(x)$.

A Proposição 9 garante que apesar de $\mathcal{F}^M$ possuir dimensionalidade arbitrariamente grande, cada vetor do conjunto $\{\omega_i^*\}_{i=1,\cdots,N}$ que minimiza o problema (2.84) é expresso como uma combinação linear finita do núcleo K associado ao espaço $\mathcal{F}^M$, a saber

$$\omega_i^*(\cdot) = \sum_{j=1}^{m_i} \alpha_{ij}^* K(\cdot, x_{ij}) + \sum_{j=1}^{p} \beta_{ij}^* K(\cdot, \chi_j), \tag{2.86}$$

em que α_{ij}^* e β_{ij}^* são grandezas escalares. Ademais, como K é o núcleo associado a $\mathcal{F}^M$ e $\omega_i \in \mathcal{F}^M$, para qualquer $x \in \mathbb{R}^n$, tem-se, pela Definição 11, que

$$f_i(x) = \langle \omega_i, K(\cdot, x) \rangle_{\mathcal{F}^M} = \omega_i(x),$$

ou seja, $f_i = \omega_i$. Dessa forma, a função $g_i^*(x)$ definida por cada nó i através da solução ω_i^* do problema (2.84) pode ser apresentada como

$$g_i(x) = \sum_{j=1}^{m_i} \alpha_{ij}^* K(x, x_{ij}) + \sum_{j=1}^{p} \beta_{ij}^* K(x, \chi_j) + b_i^*. \tag{2.87}$$

Proposição 9. *O problema* (2.84) *possui solução dada por*

$$\omega_i^*(\cdot) = \sum_{j=1}^{m_i} \alpha_{ij}^* K(\cdot, x_{ij}) + \sum_{j=1}^{p} \beta_{ij}^* K(\cdot, \chi_j) \in \mathcal{F}^M$$
$$b_i^* \in \mathbb{R}.$$

Demonstração. Basta mostra que o problema (2.84) se enquadra na classe de problemas de interesse ao teorema do representante semi-paramétrico (Teorema 4). Com efeito, utilizando as restrições de desigualdade apresentadas em (2.84), é possível

reescrever os erros ε_{ij} como $\max\{0, 1 - y_{ij}(f(x_{ij}) + b_i)\}$, de forma que o problema (2.84) pode ser reescrito como

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i=1}^{N} ||f_i||^2 + C\sum_{i=1}^{N}\sum_{j=1}^{m_i} \max\{0, 1 - y_{ij}(f(x_{ij}) + b_i)\} \\ \text{s.a.} \quad & f_i(\chi_t) - f_k(\chi_t) = 0 \qquad i = 1, \cdots, N \quad k \in \mathbb{K}_i \quad t = 1, \cdots, p \\ & b_i - b_k = 0 \qquad i = 1, \cdots, N \quad k \in \mathbb{K}_i \\ & (f_i, b_i) \in \mathcal{F}^M \times \mathbb{R} \qquad i = 1, \cdots, N, \end{aligned} \tag{2.88}$$

em que foram utilizadas a relação $f_i = \omega_i$ e as definições de $\Phi_{X_i}\omega_i$, $\Phi_\chi \omega_i$ e $\mathcal{E}_i$, esta última definida durante a derivação do *SVM* linear distribuído (2.31). Reescrito o problema, note que o lagrangeano derivado de (2.88) é estritamente convexo. Dessa forma, pela Proposição 3 é possível resgatar a solução primal a partir da solução dual realizando a minimização do lagrangeano avaliado na solução dual, a saber

$$\{f_i^*, b_i^*\}_{i=1,\cdots,N} \in \arg\min_{(f_i,b_i)\in\mathcal{F}^M\times\mathbb{R}} \mathcal{L}\left(\{f_i, \zeta_{ikt}^*, \lambda_{ik}^*\}_{\substack{i=1,\cdots,N \\ k\in\mathbb{K}_i \\ t=1,\cdots,p}}\right), \tag{2.89}$$

em que

$$\begin{aligned} \mathcal{L}\left(\{f_i, \zeta_{ikt}^*, \lambda_{ik}^*\}_{\substack{i=1,\cdots,N \\ k\in\mathbb{K}_i \\ t=1,\cdots,p}}\right) = \; & \frac{1}{2}\sum_{i=1}^{N} ||f_i||^2 + C\sum_{i=1}^{N}\sum_{j=1}^{m_i} \max\{0, 1 - y_{ij}(f(x_{ij}) + b_i)\} \\ & + \sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}\sum_{t=0}^{p} \zeta_{ikt}^*(f_i(\chi_t) - f_k(\chi_t)) + \sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i} \lambda_{ik}^*(b_i - b_k) \end{aligned}$$

com $\{(\zeta_{ikt}^*, \lambda_{ik}^*)\}_{\substack{i=1,\cdots,N \\ k\in\mathbb{K}_i \\ t=1,\cdots,p}}$ sendo a solução dual de (2.88).

Por fim, observe que, assim como realizado em (2.56), as somas com respeito as restrições apresentadas pelo lagrangeano podem ser reordenadas da seguinte forma

$$\sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}\sum_{t=0}^{p} \zeta_{ikt}^*(f_i(\chi_t) - f_k(\chi_t)) = \sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}\sum_{t=0}^{p} f_i(\chi_t)(\zeta_{ikt}^* - \zeta_{kit}^*) \tag{2.90}$$

$$\sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i} \lambda_{ik}(b_i - b_k) = \sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i} b_i(\lambda_{ik}^* - \lambda_{ki}^*). \tag{2.91}$$

Assim, o problema (2.89) pode ser separado entre os nós da rede, de forma que a minimização a cada nó pode ser reescrita como

$$(f_i^*, b_i^*) \in \arg\min_{\substack{f_i\in\mathcal{F}^M \\ b_i\in\mathbb{R}}} \theta(||f||) + \gamma\left(\left\{(x_{ij}, y_{ij}, \chi_t, f_i(x_{ij}) + b_i, f_i(\chi_t), b_i)\right\}_{\substack{j=1,\cdots,m_i \\ t=1,\cdots,p}}\right), \tag{2.92}$$

em que

$$\begin{aligned}\theta &= \frac{1}{2}\sum_{i=1}^{N}||f_i||^2 \\ \gamma &= C\sum_{i=1}^{N}\sum_{j=1}^{m_i}\max\{0,1-y_{ij}(f(x_{ij})+b_i)\}+\sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}\sum_{t=0}^{p}f_i(\chi_t)(\zeta^*_{ikt}-\zeta^*_{kit}) \\ &\quad +\sum_{i=1}^{N}\sum_{k\in\mathbb{K}_i}b_i(\lambda^*_{ik}-\lambda^*_{ki}),\end{aligned}$$

o qual claramente se enquadra na classe de problemas do Teorema 4. Portanto,

$$f_i^*(\cdot)=\omega_i^*(\cdot)=\sum_{j=1}^{m_i}\alpha^*_{ij}K(\cdot,x_{ij})+\sum_{j=1}^{p}\beta^*_{ij}K(\cdot,\chi_j)\in\mathcal{F}^M$$
$$b_i^*\in\mathbb{R}$$

■

2.4.2.1 Formulação do Algoritmo Distribuído

Ratificando que a ideia de consenso entre os agentes da rede se limita a impor que os nós concordem entre si quanto a transformação linear determinada por Φ_χ com imagem de dimensão $p < M$, claramente o problema (2.84) deve encontrar os coeficientes α^*_{ij},β^*_{it} e b_i^* introduzidos em (2.87) de forma a garantir esse consenso entre os agentes. Contudo para determiná-los de uma forma distribuída é necessário, assim como no caso linear, aplicar o método *ADMM* ao problema (2.84).

A priori, como realizado em (2.31), são introduzidas variáveis artificiais para desacoplar as restrições consensuais, a saber,

$$\begin{aligned}&\Phi_\chi\omega_i-u_{ik}=0 \quad i=1,\cdots,N \quad k\in\mathbb{K}_i\\ &\Phi_\chi\omega_k-u_{ik}=0 \quad i=1,\cdots,N \quad k\in\mathbb{K}_i\\ &b_i-h_{ik}=0 \quad i=1,\cdots,N \quad k\in\mathbb{K}_i\\ &b_k-h_{ik}=0 \quad i=1,\cdots,N \quad k\in\mathbb{K}_i,\end{aligned}$$

em que $u_{ik}\in\mathbb{R}^p$ e $h_{ik}\in\mathbb{R}$. Assim, é possível reescrever (2.84) de forma a se enquadrar na classe de problemas de interesse ao *ADMM* (1.37). A saber, de forma análoga a derivação de (2.44), tem-se que (2.84) pode ser reescrito como

$$\begin{aligned}\min\quad & F(v)\\ \text{s.a.}\quad & Av-B\kappa=0\\ & v\in\mathcal{F}^{NM}\times\mathbb{R}\\ & \kappa\in\mathbb{R}^{2E(p+1)},\end{aligned}\tag{2.93}$$

em que $v=[\omega \quad b]^T$, $\kappa=[u \quad h]^T$, E é o número de arestas do grafo que representa a rede de interesse e

$$F(v)=\sum_{i=0}^{N}\frac{1}{2}||\omega_i||^2+C\sum_{i=1}^{N}1_i^T\max\{0,1-Y_i(\Phi_{X_i}\omega_i+b_i1_i)\}.$$

Ademais, define-se

$$\omega = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_N \end{bmatrix} \in \mathcal{F}^{NM} \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} \in \mathbb{R}^N$$

$$u = \begin{bmatrix} \{u_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{u_{Nk}\}_{k\in\mathbb{K}_N} \end{bmatrix} \in \mathbb{R}^{2Ep} \qquad h = \begin{bmatrix} \{h_{1k}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{h_{Nk}\}_{k\in\mathbb{K}_N} \end{bmatrix} \in \mathbb{R}^{2E}$$

$$A = \begin{bmatrix} \hat{A} & 0 \\ \hat{A} & 0 \\ 0 & \tilde{A} \\ \underbrace{0}_{\mathcal{F}^{4E(p+1)\times NM}} & \underbrace{\tilde{A}}_{\mathbb{R}^{4E(p+1)\times N}} \end{bmatrix} \qquad B = \begin{bmatrix} I & 0 \\ P & 0 \\ 0 & I \\ 0 & T \end{bmatrix} \in \mathbb{R}^{4E(p+1)\times 4E(p+1)},$$

com

$$\begin{aligned} \hat{A} &= \operatorname{diag}(\hat{A}_1, \cdots, \hat{A}_N) \in \mathcal{F}^{2Ep\times NM} \\ \tilde{A} &= \operatorname{diag}(\tilde{A}_1, \cdots, \tilde{A}_N) \in \mathbb{R}^{2E\times N}. \end{aligned}$$

Por sua vez, as matrizes P e T foram definidas em (2.40), contudo nesse caso o produto de *Kronecker* é feito com relação a matriz identidade em $\mathbb{R}^{p\times p}$. Por fim, os vetores $\hat{A}_i$ e $\tilde{A}_i$ são definidos como

$$\hat{A}_i = \begin{bmatrix} \Phi_\chi \\ \vdots \\ \Phi_\chi \end{bmatrix} \in \mathcal{F}^{\#\mathbb{K}_i p\times M} \qquad \tilde{A}_i = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{\#\mathbb{K}_i}$$

Assim, é possível aplicar o método *ADMM* ao problema (2.93) de forma a obter

$$\begin{aligned} \nu^{l+1} &\in \arg\min_{\nu\in\mathcal{F}^{N+M}\times\mathbb{R}} \mathcal{L}_c(\nu, \kappa^l, \lambda^l) \\ \kappa^{l+1} &\in \arg\min_{\kappa\in\mathbb{R}^{2E(n+1)}} \mathcal{L}_c(\nu^{l+1}, \kappa, \lambda^l) \\ \lambda^{l+1} &= \lambda^l + c(A\nu^{l+1} - B\kappa^{l+1}), \end{aligned} \tag{2.94}$$

em que $\mathcal{L}_c$ é o lagrangeano aumentado pertinente a (2.93), a saber,

$$\mathcal{L}_c(\nu, \kappa, \lambda) = F(\nu) + (A\nu - B\kappa)^T\lambda + \frac{c}{2}||A\nu - B\kappa||^2. \tag{2.95}$$

De forma análoga a derivação do Algoritmo 6, são abandonadas as notações condensadas para resgatar os somatórios e, assim, construir um algoritmo a partir de (2.94) que possa ser distribuído. Com efeito, são introduzidas notações para diferentes blocos que integram a variável dual de forma a comportar as $4E(p+1)$ restrições

pertinentes ao problema, a saber, toma-se

$$\lambda = \begin{bmatrix} \{\lambda_{1k_1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\lambda_{Nk_1}\}_{k\in\mathbb{K}_N} \\ \{\lambda_{k1_2}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\lambda_{kN_2}\}_{k\in\mathbb{K}_N} \\ \{\zeta_{1k_1}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\zeta_{Nk_1}\}_{k\in\mathbb{K}_N} \\ \{\zeta_{k1_2}\}_{k\in\mathbb{K}_1} \\ \vdots \\ \{\zeta_{kN_2}\}_{k\in\mathbb{K}_N} \end{bmatrix}, \tag{2.96}$$

em que $\lambda_{ik_1}, \lambda_{ki_2} \in \mathbb{R}^p$ e $\zeta_{ik_1}, \zeta_{ki_2} \in \mathbb{R}$. Reintroduzindo em (2.94) a variável $\mathcal{E}_i$ e utilizando a definição em (2.96) é possível derivar que

$$\begin{aligned} (\nu,\mathcal{E})^{l+1} &\in \arg\min_{(\nu,\mathcal{E})\in\mathbb{H}} \mathcal{L}_c\Big((\nu,\mathcal{E}),\kappa^l,\lambda^l\Big) \\ \kappa^{l+1} &\in \arg\min_{\kappa\in\mathbb{R}^{2E(p+1)}} \mathcal{L}_c\Big((\nu,\mathcal{E})^{l+1},\kappa,\lambda^l\Big) \\ \lambda^{l+1} &= \lambda^l + c(Av^{l+1} - Bu^{l+1}), \end{aligned} \tag{2.97}$$

em que

$$\begin{aligned} &\mathcal{L}_c\Big((\nu,\mathcal{E}),\kappa^l,\lambda^l\Big) = \\ &\sum_{i=0}^{N}\Bigg\{\frac{1}{2}||\omega_i||^2 + C\sum_{i=1}^{N}1_i^T\mathcal{E}_i + \sum_{k\in\mathbb{K}_i}\Big[(\Phi_\chi\omega_i - u_{ik})^T\lambda_{ik_1} + (\Phi_\chi\omega_i - u_{ki})^T\lambda_{ki_2} \\ &+(b_i - h_{ik})\zeta_{ik_1} + (b_i - h_{ki})\zeta_{ki_2} + \frac{c}{2}||\Phi_\chi\omega_i - u_{ik}||^2 \\ &+\frac{c}{2}||\Phi_\chi\omega_i - u_{ki}||^2 + \frac{c}{2}||b_i - h_{ik}||^2 + \frac{c}{2}||b_i - h_{ki}||^2\Big]\Bigg\}. \end{aligned} \tag{2.98}$$

Ademais, define-se que

$$\mathbb{H} = \{(\nu,\mathcal{E}) \mid (\nu_i,\mathcal{E}_i) \in \mathbb{H}_i \quad i = 1,\cdots,N\}$$

com $\mathbb{H}_i = \big\{(\nu_i,\mathcal{E}_i) \mid 1 - Y_i(\Phi_{X_i}\omega_i + b_i 1_i) - \mathcal{E}_i \leqslant 0 \text{ e } -\mathcal{E}_i \leqslant 0\big\}$.

Agora, analogamente ao Algoritmo 6, observe que a minimização com respeito a variável $(\nu,\mathcal{E})$ é separável entre as variáveis $\{(\omega_i,b_i,\mathcal{E}_i)\}_{i=1,\cdots,N}$. A saber, realiza-se a cada nó i a seguinte minimização

$$(\omega_i,b_i,\mathcal{E}_i)^{l+1} \in \arg\min_{(\omega_i,b_i\mathcal{E}_i)\in\mathbb{H}_i} \mathcal{P}_c\Big((\omega_i,b_i\mathcal{E}_i),\{u_{ik}^l,h_{ik}^l,\lambda_{ik_1}^l,\zeta_{ik_1}^l,u_{ki}^l,h_{ki}^l,\lambda_{ki_2}^l,\zeta_{ki_2}^l\}_{k\in\mathbb{K}_i}\Big)$$

com

$$\begin{aligned}\mathcal{P}_c\Big((\omega_i,b_i,\mathcal{E}_i),\{u_{ik}^l,h_{ik}^l,\lambda_{ik_1}^l,\zeta_{ik_1}^l,u_{ki}^l,h_{ki}^l,\lambda_{ki_2}^l,\zeta_{ki_2}^l\}_{k\in\mathbb{K}_i}\Big) = \\ \frac{1}{2}||\omega_i||^2 + C1_i^T\mathcal{E}_i + \sum_{k\in\mathbb{K}_i}\Big[(\Phi_\chi\omega_i)^T\lambda_{ik_1}^l + (\Phi_\chi\omega_i)^T\lambda_{ki_2}^l + b_i\zeta_{ik_1}^l + b_i\zeta_{ki_2}^l \\ +\frac{c}{2}\left(||\Phi_\chi\omega_i - u_{ik}^l||^2 + ||\Phi_\chi\omega_i - u_{ki}^l||^2 + ||b_i - h_{ik}^l||^2 + ||b_i - h_{ki}^l||^2\right)\Big].\end{aligned} \tag{2.99}$$

Ainda como no caso para a classificação linear, as minimizações com respeito as variáveis auxiliares u e h condensadas em κ podem, também, serem feitas separadamente para cada nó vizinho de um determinado nó i. A saber, cada nó i realiza o seguinte conjunto de minimizações

$$\begin{aligned}u_{ik}^{l+1} &\in \arg\min_{u_{ik}\in\mathbb{R}^p}\mathcal{U}_c\left(\omega_i^{l+1},\omega_k^{l+1},u_{ik},\lambda_{ik_1}^l,\lambda_{ik_2}^l\right) \quad \forall k\in\mathbb{K}_i \\ h_{ik}^{l+1} &\in \arg\min_{h_{ik}\in\mathbb{R}}\mathcal{H}_c\left(b_i^{l+1},b_k^{l+1},h_{ik},\zeta_{ik_1}^l,\zeta_{ik_2}^l\right) \quad \forall k\in\mathbb{K}_i\end{aligned} \tag{2.100}$$

com

$$\begin{aligned}\mathcal{U}_c\left(\omega_i^{l+1},\omega_k^{l+1},u_{ik},\lambda_{ik_1}^l,\lambda_{ik_2}^l\right) = \\ -(\lambda_{ik_1}^l)^Tu_{ik} - (\lambda_{ik_2}^l)^Tu_{ik} + \frac{c}{2}||\Phi_\chi\omega_i^{l+1} - u_{ik}||^2 + \frac{c}{2}||\Phi_\chi\omega_k^{l+1} - u_{ik}||^2 \\ \mathcal{H}_c\left(b_i^{l+1},b_k^{l+1},h_{ik},\zeta_{ik_1}^l,\zeta_{ik_2}^l\right) = \\ -(\zeta_{ik_1}^l)^Tu_{ik} - (\zeta_{ik_2}^l)^Tu_{ik} + \frac{c}{2}||b_i^{l+1} - h_{ik}||^2 + \frac{c}{2}||b_k^{l+1} - h_{ik}||^2.\end{aligned} \tag{2.101}$$

Agora, seguindo o mesmo raciocínio realizado na derivação do Algoritmo 7, é possível evitar a dependência nas variáveis auxiliares u e h e nas variáveis duais λ_{ik_2} e ζ_{ik_2}. De fato, uma vez que os problemas em (2.100) são quadráticos, a partir da condição de primeira ordem aplicada a esses problemas, da iteração sobre as variáveis duais em (2.97) e inicializando essas variáveis com o vetor nulo, deriva-se, para toda iteração l, que

$$\begin{aligned}u_{ik}^l &= u_{ki}^l \\ h_{ik}^l &= h_{ki}^l \\ \lambda_{ik}^l &= \lambda_{ik_1}^l = -\lambda_{ki_2}^l \\ \zeta_{ik}^l &= \zeta_{ik_1}^l = -\zeta_{ki_2}^l \\ \lambda_{ik}^l &= -\lambda_{ki}^l \\ \zeta_{ik}^l &= -\zeta_{ki}^l,\end{aligned}$$

em que

$$\begin{aligned}u_{ik}^l &= \frac{1}{2}(\Phi_\chi\omega_i^l + \Phi_\chi\omega_k^l) \\ h_{ik}^l &= \frac{1}{2}(b_i^l + b_k^l) \\ \lambda_{ik}^l &= \lambda_{ik}^{l-1} + \frac{c}{2}(\Phi_\chi\omega_i^l - \Phi_\chi\omega_k^l) \\ \zeta_{ik}^l &= \zeta_{ik}^{l-1} + \frac{c}{2}(b_i^l - b_k^l).\end{aligned}$$

Por fim, utilizando as identidades para as variáveis auxiliares e as relações entre as variáveis duais é possível reescrever (2.99) como

$$\begin{aligned}&\mathcal{P}_c\Big((\omega_i,b_i,\mathcal{E}_i),(\omega_i,b_i)^l,\{(\omega_k,b_k)^l\}_{k\in\mathbb{K}_i},\lambda_i^l,\zeta_i^l\Big)=\\&\frac{1}{2}||\omega_i||^2+C1_i^T\mathcal{E}_i+2(\Phi_\chi\omega_i)^T\lambda_i^l+2b_i\zeta_i^l\\&+c\sum_{k\in\mathbb{K}_i}\left(||\Phi_\chi\omega_i-\frac{1}{2}(\Phi_\chi\omega_i^l+\Phi_\chi\omega_k^l)||^2+||b_i-\frac{1}{2}(b_i^l+b_k^l)||^2\right).\end{aligned}\tag{2.102}$$

em que $\zeta_i=\sum_{k\in\mathbb{K}_i}\zeta_{ik}$ e $\lambda_i=\sum_{k\in\mathbb{K}_i}\lambda_{ik}$. Dessa forma, é possível afirmar que cada nó i computa a cada iteração o seguinte conjunto de problemas

$$(\omega_i,b_i,\mathcal{E}_i)^{l+1}\in\arg\inf_{(\omega_i,b_i\mathcal{E}_i)\in\mathbb{H}_i}\mathcal{P}_c\Big((\omega_i,b_i,\mathcal{E}_i),(\omega_i,b_i)^l,\{(\omega_k,b_k)^l\}_{k\in\mathbb{K}_i},\lambda_i^l,\zeta_i^l\Big)\tag{2.103}$$

$$\lambda_i^{l+1}=\lambda_i^l+\frac{c}{2}\sum_{i\in\mathbb{K}_i}\Phi_\chi(\omega_i^{l+1}-\omega_k^{l+1})\tag{2.104}$$

$$\zeta_i^{l+1}=\zeta_i^l+\frac{c}{2}\sum_{i\in\mathbb{K}_i}(b_i^{l+1}-b_k^{l+1}).\tag{2.105}$$

Apesar do problema ter sido derivado de forma a ser distribuído, ainda há a necessidade de manipulação do vetor $\omega_i\in\mathcal{F}^M$. Entretanto, como não há mais a imposição do consenso com relação aos modelos preditivos determinados a cada agente da rede, o que importa ao problema são os classificadores em si, ou seja, a função classificação $g_i(x)$, a qual é expressa em função do núcleo (Proposição 9).

Dessa forma, busca-se determinar uma forma fechada para os coeficientes $\{\alpha_{ij}\}_{j=1,\cdots,m_i}$ e $\{\beta_{ij}\}_{j=1,\cdots,p}$ introduzidos em (2.86) bem como uma forma distribuída para determiná-los. Com efeito, de forma análoga a derivação do Algoritmo 8, aplica-se as condições *KKT* ao problema (2.103), o qual possui lagrangeano dado por

$$\begin{aligned}&\mathcal{L}\Big((\omega_i,b_i,\mathcal{E}_i),(\omega_i,b_i)^l,\{(\omega_k,b_k)^l\}_{k\in\mathbb{K}_i},\lambda_i^l,\zeta_i^l,\mu_i,\eta_i\Big)=\\&\frac{1}{2}||\omega_i||^2+C1_i^T\mathcal{E}_i+2(\Phi_\chi\omega_i)^T\lambda_i^l+2b_i\zeta_i^l\\&+c\sum_{k\in\mathbb{K}_i}\left(||\Phi_\chi\omega_i-\frac{1}{2}(\Phi_\chi\omega_i^l+\Phi_\chi\omega_k^l)||^2+||b_i-\frac{1}{2}(b_i^l+b_k^l)||^2\right)\\&+\mu_i^T(1_i-\Upsilon_i(\Phi_{X_i}\omega_i+b_i1_i)-\mathcal{E}_i)-\eta_i^T\mathcal{E}_i,\end{aligned}\tag{2.106}$$

e é convexo com restrições lineares. Então, o vetor $(\omega_i,b_i,\mathcal{E}_i)^{l+1}$ satisfaz o problema (2.103) se somente se $(\omega_i,b_i,\mathcal{E}_i)^{l+1}$ e $(\mu_i,\eta_i)^{l+1}$ satisfazem as condições *KKT*.

Dessa forma, das condições

$$\nabla_{\omega_i}\mathcal{L}\Big((\omega_i,b_i,\mathcal{E}_i)^{l+1},(\omega_i,b_i)^l,\{(\omega_k,b_k)^l\}_{k\in\mathbb{K}_i},\lambda_i^l,\zeta_i^l,\mu_i^{l+1},\eta_i^{l+1}\Big)=0\tag{2.107}$$

$$\nabla_{b_i}\mathcal{L}\Big((\omega_i,b_i,\mathcal{E}_i)^{l+1},(\omega_i,b_i)^l,\{(\omega_k,b_k)^l\}_{k\in\mathbb{K}_i},\lambda_i^l,\zeta_i^l,\mu_i^{l+1},\eta_i^{l+1}\Big)=0\tag{2.108}$$

é possível derivar que

$$\omega_i^{l+1} = G_i^{-1}\left(\Phi_{X_i}^T Y_i \mu_i^{l+1} - \Phi_\chi^T r_i^l\right) \tag{2.109}$$

$$b_i^{l+1} = \frac{1}{2c\#\mathbb{K}_i}(1_i^T Y_i \mu_i^{l+1} - s_i^l) \tag{2.110}$$

em que

$$\begin{aligned} G_i &= I - 2c\#\mathbb{K}_i \Phi_\chi^T \Phi_\chi \\ r_i^l &= 2\lambda_i^l - c\sum_{k\in\mathbb{K}_i}(\Phi_\chi \omega_i^l + \Phi_\chi \omega_k^l) \\ s_i^l &= 2\zeta_i^l - c\sum_{k\in\mathbb{K}_i}(b_i^l + b_k^l). \end{aligned}$$

Para determinar G_i^{-1} é utilizada a fórmula de *Sherman–Morrison–Woodbury*[14], a saber

$$G_i^{-1} = I - 2c\#\mathbb{K}_i \Phi_\chi^T Q_i^{-1} \Phi_\chi, \tag{2.111}$$

em que $Q_i = I + 2c\#\mathbb{K}_i \Phi_\chi \Phi_\chi^T$. Assim, substituindo essa expressão em (2.109), tem-se que

$$\omega_i^{l+1} = \Phi_{X_i}^T Y_i \mu_i^{l+1} - \Phi_\chi^T r_i^l - 2c\#\mathbb{K}_i \Phi_\chi^T Q_i^{-1} \Phi_\chi \Phi_{X_i}^T Y_i \mu_i^{l+1} + 2c\#\mathbb{K}_i \Phi_\chi^T Q_i^{-1} \Phi_\chi \Phi_\chi^T r_i^l. \tag{2.112}$$

Agora, observando que $\phi(x) = K(\cdot, x)$ e $\omega_i^l \in \mathcal{F}^M$, tem-se que

$$\omega_i^{l+1}(x) = \left\langle \phi(x), \omega_i^{l+1} \right\rangle_{\mathcal{F}^M} = \phi^T(x)\omega_i^{l+1}, \tag{2.113}$$

uma vez que K é o núcleo associado a $\mathcal{F}^M$ (Definição 11). Dessa forma, é possível escrever que

$$\begin{aligned} &\phi(x)^T \omega_i^{l+1} = \\ &\quad K(x, X_i)^T Y_i \mu_i^{l+1} - K(x,\chi)^T r_i^l - 2c\#\mathbb{K}_i K(x,\chi)^T Q_i^{-1} K(\chi, X_i) Y_i \mu_i^{l+1} \\ &\quad + 2c\#\mathbb{K}_i K(x,\chi)^T Q_i^{-1} K(\chi,\chi) r_i^l, \end{aligned} \tag{2.114}$$

ou seja,

$$\begin{aligned} &\phi(x)^T \omega_i^{l+1} = \\ &\quad K(x, X_i)^T Y_i \mu_i^{l+1} + K(x,\chi)^T \left[2c\#\mathbb{K}_i Q_i^{-1}\left(K(\chi,\chi) r_i^l - K(\chi, X_i) Y_i \mu_i^{l+1}\right) - r_i^l\right], \end{aligned} \tag{2.115}$$

[14] Apresentada na seção 3.8 de Meyer (2000).

em que

$$\begin{aligned}
\phi(x)^T\Phi_{X_i}^T &= K(x,X_i) = \left[K(x,x_{i1}),\cdots,K(x,x_{im_i})\right]^T\\
\phi(x)^T\Phi_{\chi}^T &= K(x,\chi) = \left[K(x,\chi_1),\cdots,K(x,\chi_p)\right]^T\\
\Phi_{X_i}\Phi_{X_i}^T &= K(X_i,X_i) = \begin{bmatrix} K(x_{i1},x_{i1}) & \cdots & K(x_{i1},x_{im_i})\\ \vdots & \vdots & \vdots\\ K(x_{im_i},x_{i1}) & \cdots & K(x_{im_i},x_{im_i})\end{bmatrix}\\
\Phi_{\chi}\Phi_{\chi}^T &= K(\chi,\chi) = \begin{bmatrix} K(\chi_1,\chi_1) & \cdots & K(\chi_1,\chi_p)\\ \vdots & \vdots & \vdots\\ K(\chi_p,\chi_1) & \cdots & K(\chi_p,\chi_p)\end{bmatrix}\\
\Phi_{\chi}\Phi_{X_i}^T &= K(\chi,X_i) = \begin{bmatrix} K(\chi_1,x_{i1}) & \cdots & K(\chi_1,x_{im_i})\\ \vdots & \vdots & \vdots\\ K(\chi_p,x_{i1}) & \cdots & K(\chi_p,x_{im_i})\end{bmatrix}.
\end{aligned}$$

Ainda, utilizando essa notação observe que $Q_i = I + 2c\#\mathbb{K}_i K(\chi,\chi)$ e a expressão (2.86) pode ser reescrita como

$$\omega_i^{l+1}(x) = K(x,X_i)^T\alpha_i^{l+1} + K(x,\chi)^T\beta_i^{l+1}, \tag{2.116}$$

em que $\alpha_i^{l+1} = [\alpha_{i1}^{l+1},\cdots,\alpha_{im_i}^{l+1}]^T$ e $\beta_i^{l+1} = [\beta_{i1}^{l+1},\cdots,\beta_{ip}^{l+1}]^T$. Logo, de (2.115) e (2.116), tem-se que

$$\alpha_i^{l+1} = Y_i\mu_i^{l+1} \tag{2.117}$$

$$\beta_i^{l+1} = 2c\#\mathbb{K}_i Q_i^{-1}\left(K(\chi,\chi)r_i^l - K(\chi,X_i)Y_i\mu_i^{l+1}\right) - r_i^l. \tag{2.118}$$

Para determinar μ_i^{l+1}, realiza-se o mesmo raciocínio aplicado para derivar a expressão (2.83). A saber, utilizando as condições de folga complementares impostas pelas condições *KKT* com relação ao problema (2.103), tem-se que

$$\begin{aligned}
\mu_i^{l+1} \in \arg\max_{0\leqslant\mu_i\leqslant C1_i}\Bigg\{ &-\frac{1}{2}\mu_i^T Y_i\left(\Phi_{X_i}G_i^{-1}\Phi_{X_i}^T + \frac{1_i1_i^T}{2c\#\mathbb{K}_i}\right)Y_i\mu_i\\
&+\left(1_i + Y_i\Phi_{X_i}G_i^{-1}\Phi_{\chi}^T r_i^l + \frac{s_i^l}{2c\#\mathbb{K}_i}Y_i1_i\right)^T\mu_i\Bigg\}.
\end{aligned} \tag{2.119}$$

Agora, utilizando a expressão para G_i^{-1} (2.111), tem-se que

$$\Phi_{X_i}G_i^{-1}\Phi_{X_i}^T = K(X_i,X_i) - 2c\#\mathbb{K}_i K(X_i,\chi)Q_i^{-1}K(\chi,X_i) \tag{2.120}$$

$$\Phi_{X_i}G_i^{-1}\Phi_{\chi}^T r_i^l = \left(K(X_i,\chi) - 2c\#\mathbb{K}_i K(X_i,\chi)Q^{-1}K(\chi,\chi)\right)r_i^l, \tag{2.121}$$

em que

$$\Phi_{X_i}\Phi_{\chi}^T = K(X_i,\chi) = \begin{bmatrix} K(x_{i1},\chi_1) & \cdots & K(x_{i1},\chi_p)\\ \vdots & \vdots & \vdots\\ K(x_{im_i},\chi_1) & \cdots & K(x_{im_i},\chi_p)\end{bmatrix}.$$

Substituindo essas expressões em (2.119), tem-se que

$$\mu_i^{l+1} \in \arg\max_{0\leqslant\mu_i\leqslant C1_i} \left\{ -\frac{1}{2}\mu_i^T Y_i \left(K(X_i,X_i) - \tilde{K}(X_i,X_i) + \frac{1_i 1_i^T}{2c\#\mathbb{K}_i} \right) Y_i\mu_i + \left[1_i + Y_i \left(K(X_i,\chi) - \tilde{K}(X_i,\chi) \right) r_i^l + \frac{s_i^l}{2c\#\mathbb{K}_i} Y_i 1_i \right]^T \mu_i \right\}, \tag{2.122}$$

com $\tilde{K}(Z,Z') = 2c\#\mathbb{K}_i K(Z,\chi) Q_i^{-1} K(\chi,Z')$. Assim, determinando a variável dual μ_i, é possível computar os coeficientes α_i e β_i como apresentado em (2.117) e (2.118). Entretanto, α_i e β_i dependem de ω_i, uma vez que r_i^l depende dessa variável, fato que exige a manipulação de vetores em $\mathcal{F}^M$. Para evitar esse problema, é definida uma nova variável, a saber

$$\tilde{\omega_i}^l = \Phi_\chi \omega_i^l \in \mathbb{R}^p. \tag{2.123}$$

Com isso, r_i^l passa a depender unicamente dessa nova variável. Por fim, para determinar como computar $\tilde{\omega_i}^{l+1}$ em função de μ_i^{l+1}, basta multiplicar a expressão (2.109) por Φ_χ. De fato,

$$\tilde{\omega_i}^{l+1} = \Phi_\chi G_i^{-1} \Phi_{X_i}^T Y_i \mu_i^{l+1} - \Phi_\chi G_i^{-1} \Phi_\chi^T r_i^l. \tag{2.124}$$

Novamente utilizando a expressão (2.111), é possível derivar que

$$\Phi_\chi G_i^{-1} \Phi_{X_i}^T = K(\chi,X_i) - 2c\#\mathbb{K}_i K(\chi,\chi) Q_i^{-1} K(\chi,X_i) \tag{2.125}$$

$$\Phi_\chi G_i^{-1} \Phi_\chi^T = K(\chi,\chi) - 2c\#\mathbb{K}_i K(\chi,\chi) Q_i^{-1} K(\chi,\chi). \tag{2.126}$$

Substituindo essas expressões em (2.124), tem-se que

$$\tilde{\omega_i}^{l+1} = \left(K(\chi,X_i) - \tilde{K}(\chi,X_i) \right) Y_i \mu_i^{l+1} - \left(K(\chi,\chi) - \tilde{K}(\chi,\chi) \right) r_i^l, \tag{2.127}$$

em que $r_i^l = 2\lambda_i^l - c \sum_{i\in\mathbb{K}_i} (\tilde{\omega_i}^l + \tilde{\omega_k}^l)$.

Enfim, utilizando as expressões (2.122), (2.127), (2.110) (2.104) e (2.105), é possível determinar o Algoritmo 9, o qual soluciona o problema de suporte de máquina vetorial para o caso não linear.

Observe que a matriz Q_i sempre possui inversa, uma vez que a matriz $K(\chi,\chi)$ é semi-definida positiva. Ademais, a cada iteração do Algoritmo 9 é possível determinar a cada nó uma função para classificação local de um dado vetor $x \in \mathbb{R}^n$ através da expressão

$$g_i^{l+1}(x) = K(x,X_i)^T \alpha_i^{l+1} + K(x,\chi)^T \beta_i^{l+1} + b_i^{l+1}, \tag{2.128}$$

em que seu sinal determina a classe y associada a x. Ademais, os vetores α_i^{l+1} e β_i^{l+1} são determinados respectivamente pelas expressões (2.117) e (2.118).

Algoritmo 9 – SVM Não Linear Distribuído (NDSVM)

Dados: $\lambda_i^0 = 0$, $\zeta_i^0 = 0$ e ${\tilde{\omega}_i}^0$ para todo $i = 1, \cdots, N$
para cada *nó i* **faça**
para $l = 0, 1, 2, 3, \ldots$ **faça**

$$
\begin{aligned}
\mu_i^{l+1} \in \arg \max_{0 \leqslant \mu_i \leqslant C1_i} \Bigg\{ & -\frac{1}{2}\mu_i^T Y_i \left(K(X_i, X_i) - \tilde{K}(X_i, X_i) + \frac{1_i 1_i^T}{2c\#\mathbb{K}_i} \right) Y_i \mu_i \\
& + \left[1_i + Y_i \left(K(X_i, \chi) - \tilde{K}(X_i, \chi) \right) r_i^l + \frac{s_i^l}{2c\#\mathbb{K}_i} Y_i 1_i \right]^T \mu_i \Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
{\tilde{\omega}_i}^{l+1} &= \left(K(\chi, X_i) - \tilde{K}(\chi, X_i) \right) Y_i \mu_i^{l+1} - \left(K(\chi, \chi) - \tilde{K}(\chi, \chi) \right) r_i^l \\
b_i^{l+1} &= \frac{1}{2c\#\mathbb{K}_i} (1_i^T Y_i \mu_i^{l+1} - s_i^l) \\
\lambda_i^{l+1} &= \lambda_i^l + \frac{c}{2} \sum_{i \in \mathbb{K}_i} {\tilde{\omega}_i}^{l+1} - {\tilde{\omega}_k}^{l+1}) \\
\zeta_i^{l+1} &= \zeta_i^l + \frac{c}{2} \sum_{i \in \mathbb{K}_i} (b_i^{l+1} - b_k^{l+1})
\end{aligned}
$$

com

$$
\begin{aligned}
r_i^l &= 2\lambda_i^l - c \sum_{i \in \mathbb{K}_i} ({\tilde{\omega}_i}^l + {\tilde{\omega}_k}^l) \\
s_i^l &= 2\zeta_i^l - c \sum_{k \in \mathbb{K}_i} (b_i^l + b_k^l) \\
Q_i &= I + 2c\#\mathbb{K}_i K(\chi, \chi)
\end{aligned}
$$

fim
fim

II

Simulações

CAPÍTULO 3

Aplicação Numérica

Apresentação de simulações numéricas aplicadas à máquina de suporte vetorial distribuída linear e não linear sobre uma rede conexa gerada de forma aleatória.

3.1 IMPLEMENTAÇÃO

A linguagem utilizada para a implementação do Algoritmo 8 foi o *Python*. Esta linguagem é eficaz para simulação numérica por possuir diversos arcabouços relevantes ao tratamento, visualização e análise de dados. Alguns arcabouços pertinentes são pandas (versão 0.20.2), numpy (versão 1.13.0), scikit-learn (versão 0.18), matplotlib (versão 2.0.1) e seaborn (versão 0.7.1).

Para que os testes pudessem ser feitos de forma abrangente e flexível, procurou-se implementar não somente o algoritmo proposto, mas, também, um ambiente adequado que simulasse a situação desejada, uma vez que não se dispunha de uma rede real de computadores. Dessa forma, implementou-se um programa tal que, a partir do número de nós desejado, cria uma rede conexa e descentralizada distribuída aleatoriamente.

Quanto aos dados de treino, foram utilizados dados de classificação binária gerados artificialmente e, também, dados provenientes de aplicações reais. Com respeito aos dados reais, houve a necessidade de realizar o pré-processamento dos mesmos de forma a evitar entradas com atributos faltantes (através da imputação pela média[1]) e padronizar as classes de forma a estarem rotuladas entre os inteiros 1 e -1, assim como exigido pela teoria (Capítulo 2). Ainda, para evitar a diferença exacerbada entre os valores de cada atributo[2], procura-se torná-los adimensionais. Para tanto, cada atributo é normalizado de forma a possuir média zero e desvio padrão unitário.

[1] Processo em que cada coordenada faltante de um vetor é substituída pela média dos valores atribuídos à essa coordenada pelos demais vetores do conjunto de dados, ou seja, os atributos faltantes são substituídos pela média dos valores da coluna que referencia esse mesmo atributo.

[2] Isso é necessário para evitar a instabilidade numérica e o impacto dos atributos com valores maiores em detrimento dos atributos com valores menores durante a computação do núcleo avaliado no conjunto de dados [Seção 2.2 de Hsu, Chang e Lin (2010)].

Ademais, os parâmetros dos modelos distribuído e centralizado foram determinados de forma empírica através da busca em grade[3]. Há processos mais rebuscados para determinar os parâmetros pertinentes à algoritmos de aprendizagem, como os oriundos do *SVM*. Por exemplo, os métodos que são apresentados em Chapelle et al. (2002) e Klatzer e Pock (2015) procuram determinar os parâmetros do *SVM* através da otimização contínua. Entretanto, determinar os parâmetros do *SVM* distribuído de forma mais elaborada é, por si só, ainda tema de pesquisa. Por isso, optou-se pelo processo tradicional de busca em grade.

Contudo, para determinar os melhores parâmetros através da busca em grade, é necessário comparar a capacidade de predição dos modelos provenientes de cada algoritmo (*SVM* centralizado e *SVM* distribuído). Para estimar a capacidade preditiva dos classificadores obtidos pelas variantes centralizada e distribuída, optou-se por utilizar a validação cruzada para k amostras estratificadas. Nesse processo o conjunto de dados é separado em k subconjuntos que possuem proporções de exemplos de cada classe semelhantes à apresentada pelo conjunto de dados total. Destes, $k-1$ subconjuntos são usados no treinamento de um dado classificador que, por sua vez, é testado no conjunto de dados restante. O processo é repetido de forma que cada um dos subconjuntos seja utilizado para teste apenas uma vez. Então, a acurácia é estimada como a média nessas k repetições. Nesse trabalho, foi utilizado k igual a 3. Por fim, a acurácia de cada modelo frente a um conjunto de teste é determinada simplesmente pela razão entre o número de predições corretas e o número total de dados do conjunto de teste. Ainda, como a rede é gerada artificialmente, os dados para o treino do algoritmo distribuído são divididos em conjuntos não necessariamente estratificados entre os nós da rede formando os dados locais de cada nó. É importante ressaltar que os conjuntos de treino e teste são normalizados segundo a média e o desvio padrão do conjunto de treino de forma a evitar a aquisição de informação prévia a respeito do conjunto de teste [seção 2.2 de Hsu, Chang e Lin (2010)].

Além disso, o gerenciamento de troca de mensagens entre os agentes da rede e da topologia desta foi feito através do arcabouço mpi4py (versão 2.0.0) que é uma versão do *OpenMPI* para a linguagem *Python*. Note que a comunicação é de suma importância, uma vez que as restrições consensuais exigem que cada nó transfira ao menos um vetor para seus vizinhos a cada nova iteração do algoritmo distribuído, de forma que tal transferência deve ser precedida de uma sincronização. Ainda, para tratar o problema de maximização frente a variável dual μ_i foi utilizado o arcabouço cvxopt (versão 1.1.9), o qual é voltado à solução de problemas de otimização convexa implementado em *Python*. Apesar dessa escolha, é importante ressaltar que caso seja necessário uma implementação robusta voltada para alta performasse é

[3] Processo que simplesmente faz uma busca exaustiva através de parâmetros determinados manualmente.

imprescindível a utilização de linguagens compiladas como *C* ou *FORTRAN*. Neste caso, assim como já comentado, o método *GENCAN* seria mais apropriado.

Por fim, para efeito de comparação, buscou-se determinar modelos provenientes do *SVM* centralizado aplicado ao conjunto local e total de dados como se os mesmos não estivessem distribuídos, ou seja, modelos que utilizam, respectivamente, como dados de treino o conjunto local e total de dados. A implementação para o *SVM* centralizado, como já discutido na subseção 2.2.3, já é bem conhecida. Portanto, utilizou-se as classe *LinearSVC* e *SVC* do arcabouçou scikit-learn (versão 0.18) para regularização l_1.

3.1.1 *Organização do Código*

O código foi implementado de forma fracionada em diferentes módulos. A saber, a Figura 4 apresenta a disposição dos arquivos pertinentes à essa implementação. Nessa hierarquia, o diretório `src/` contem os módulos implementados, a

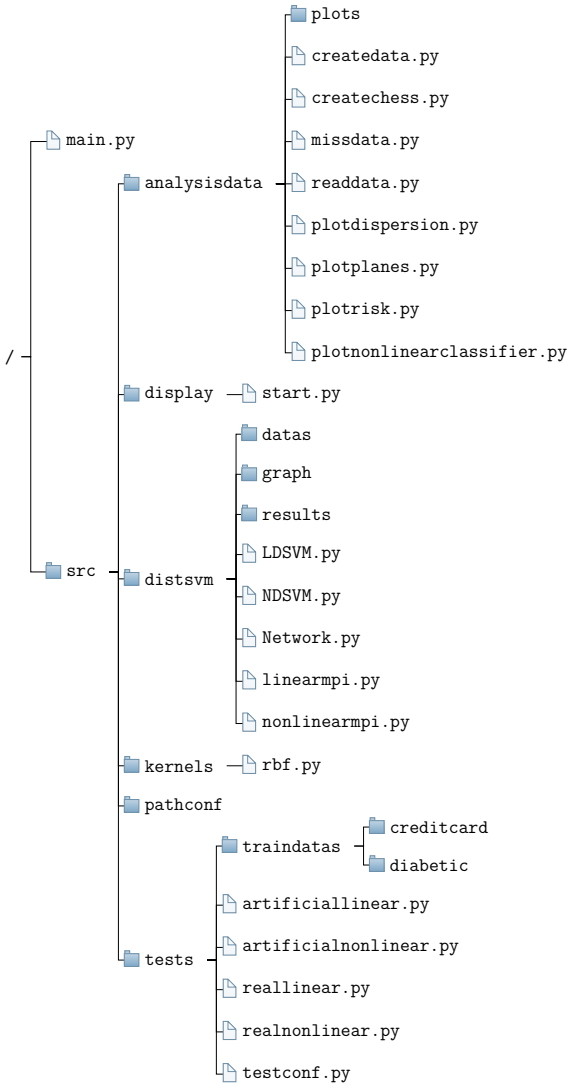


Figura 4 – Hierarquia de diretórios apresentada pelo código implementado.

saber `analysisdata/`, `display/`, `distsvm/`, `kernels`, `pathconf/` e `tests/`. Os quais são brevemente descritos abaixo,

`analysisdata/` Responsável pelo tratamento e criação de dados, além de gerar a visualização dos resultados provenientes dos testes realizados. A saber, `readdata.py` gerencia a leitura dos dados de treino para um determinado teste, `missdata.py` trata dos dados faltantes nos conjuntos de dados reais, `createdata.py` cria os dados de treino artificiais utilizados no treino do Algoritmo 8, `createchess.py` cria os dados de treino artificiais utilizados no treino do Algoritmo 9, `plotrisk.py` gera um gráfico que apresenta os riscos[4] dos modelos centralizado, local e distribuído frente ao conjunto conjunto real de dados, `plotplanes.py` gera um gráfico que apresenta os modelos provenientes do *SVM* linear distribuído e do *SVM* linear central aplicados tanto para o conjunto total quanto para o conjunto local de dados artificiais gerados por `createdata.py`, `plotdispersion.py` gera um gráfico que apresenta como a diferença entre os resultados de cada agente se comporta a cada iteração do Algoritmo 8, `plotnonlinearclassifier.py` gera gráficos que apresentam os classificadores oriundos dos métodos *SVM* não linear distribuído e central, este último aplicado ao conjunto total e local de dados artificiais gerados por `createchess.py` e, por fim, os gráficos gerados por esse módulo são armazenados no diretório `plots/`.

`display/` Responsável por criar uma interface com o usuário a partir do método em `start.py`.

`distsvm/` Responsável por configurar o ambiente e implementar o Algoritmo 8 e o Algoritmo 9. A saber, o `Network.py` é a classe que armazena os atributos e métodos pertinentes a rede, o `LDSVM.py` é a classe responsável por gerir os métodos pertinentes ao *SVM* distribuído linear como a busca em grade e a determinação da acurácia, de forma análoga o `NDSVM.py` é a classe responsável por gerir o *SVM* distribuído não linear e os arquivos `linearmpi.py` e `nonlinearmpi.py` contém os códigos que implementam, respectivamente, o Algoritmo 8 e o Algoritmo 9, os quais são processados por todos os nós da rede. Ainda, esse módulo possui os diretórios `datas/`, `graph/` e `results/`, os quais armazenam, respectivamente, os conjuntos locais de dados, as informações pertinentes a topologia da rede e os modelos determinados pelo *SVM* distribuído em ambas as variantes, linear e não linear.

`kernels/` Responsável fornecer os núcleos de interesse, nesse caso só é utilizado o núcleo *RBF* que se encontra em `rbf.py`.

[4] Note que o risco nada mais é que o evento complementar da acurácia.

`pathconf/` Responsável por gerir os caminhos nos quais são armazenados os gráficos, os dados locais, os modelos e as informações da rede.

`tests/` Responsável por implementar os testes de interesse às simulações numéricas. A saber, os arquivos com sufixo *linear* se referem aos testes vinculados ao Algoritmo 8, enquanto os arquivos com sufixo *nonlinear* se referem aos testes vinculados ao Algoritmo 9. Por outro lado, o prefixo *artificial* se refere aos testes frente ao conjunto de dados artificiais, no caso linear oriundo de `createdata.py` e no caso não linear oriundo de `createchess.py`. Já o prefixo *real* se refere aos testes frente ao conjunto de dados provenientes de um aplicação real. Esses dados de aplicações reais são encontrados em `traindata/` e são divididos em dois conjuntos de dados, a saber `creditcard/` e `diabetic/`.

3.2 ESTUDOS DE CASO À CLASSIFICAÇÃO LINEAR

Nesta seção serão apresentadas duas simulações numéricas para o Algoritmo 8. Todo o processo descrito pode ser reproduzido a partir do código implementado, descrito na seção anterior, o qual se encontra hospedado no GitHub[5].

3.2.1 *Dados Artificiais*

Os dados artificiais foram gerados a partir do método `make_classification` da classe `datasets` pertencente ao arcabouço scikit-learn (versão 0.18). Nesse caso, foram tomados 2500 dados bidimensionais com duas classes com mesmas quantidades de amostras distribuídas sobre a combinação de quatro gaussianas normalizadas. Note que a opção por dados bidimensionais se deve a facilidade de visualização dos modelos preditivos gerados durantes o teste.

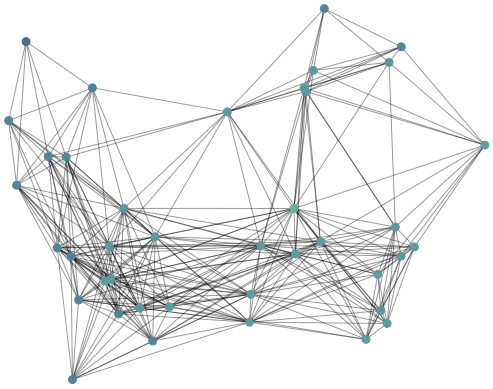


Figura 5 – Rede conexa utilizada para a simulação do Algoritmo 8 aplicado aos dados artificiais.

[5] Hospedado em https://github.com/caiodadauto/Distributed-SVM.

Dessa forma, foram treinados três modelos preditivos oriundos do *SVM* distribuído linear e do *SVM* central aplicado ao conjunto total e local de dados, esses últimos são tomados apenas para efeito de comparação. Observe que o modelo central

Tabela 2 – Parâmetros pertinentes a busca em grade à classificação linear com os dados artificiais.

SVM Distribuído		*SVM* Central
C	c	C
1	2^2	0.1
2^2	2^3	0.3
2^3	2^4	0.5
2^4	2^5	1
2^5	-	2
-	-	6
-	-	2^3

aplicado ao conjunto local de dados se refere a situação em que cada agente treina seu classificador usando somente seus dados sem a cooperação com os demais agentes da rede. Para determinar os parâmetros C e c da abordagem distribuída (Algoritmo 8) e o parâmetro C da abordagem centralizada, foi realizada uma busca em grade com respeito aos parâmetros apresentados em Tabela 2.

Determinado os parâmetros ótimos contidos nessa grade, foram estimados os riscos provenientes de cada modelo aplicado aos dados artificiais após 400 iterações. Como já comentado, essa estimativa foi realizada a partir da média das acurácias provenientes da validação cruzada para 3 amostras estratificadas. Ademais, quanto ao modelo distribuído, o risco foi determinado como a média dos riscos apresentados por cada agente da rede. A qual foi tomada de forma conexa para 40 nós com topologia determinada de forma aleatória, rede que é apresentada na Figura 5

Ainda, a Tabela 3 apresenta os riscos obtidos para o modelos distribuído, central e local provenientes dessa simulação. Esses resultados evidenciam a semelhança

Tabela 3 – Estimativa dos riscos apresentados por cada modelo aplicado aos dados artificiais após 400 iterações.

SVM Distribuído	*SVM* Central	*SVM* Local
0.0299	0.0301	0.0365

quanto a acurácia proveniente dos modelos distribuído e central. Ademais, os modelos

central, local e distribuído são apresentados na Figura 6, a qual foi gerada pelo método em `plotplanes.py`.

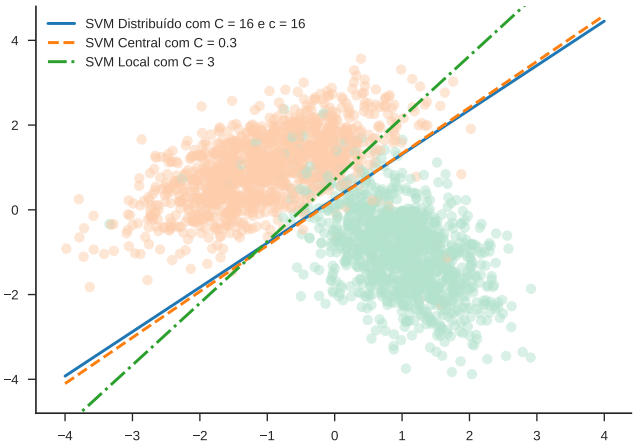


Figura 6 – Modelos gerados para o conjunto de dados artificial.

Por último, foi analisado o comportamento da diferença entre os modelos determinados a cada iteração por cada agente. Para tanto, foi definida uma função denominada por $\Delta^l(\bar{v}^l)$ que quantifica essa diferença, a saber

$$\Delta^l(\bar{v}^l) = \frac{1}{N}\sum_{i=1}^{N} ||v_i^l - \bar{v}^l||,$$

em que $\bar{v}^l = \frac{1}{N}\sum_{i=0}^{N} v_i^l$. Dessa forma, foi gerado um gráfico que apresenta $\Delta^l(\bar{v}^l)$ para as 400 primeiras iterações do método distribuído. A saber,

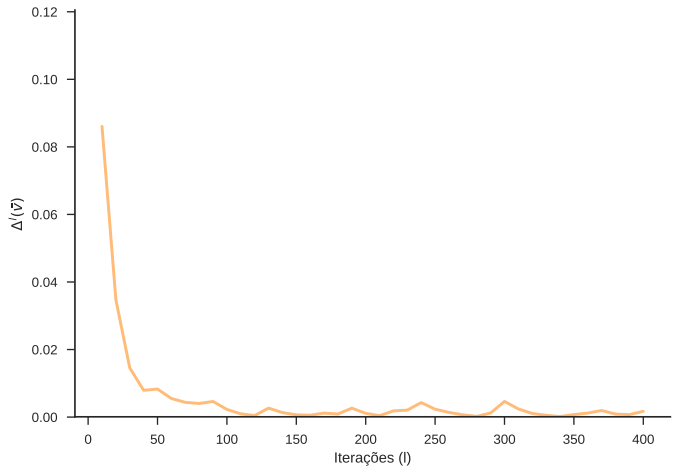


Figura 7 – Dispersão entre os modelos gerados a cada iteração para cada agente da rede.

De fato, como imposto pelas restrições consensuais, as soluções determinadas pelos agentes da rede convergem para uma solução comum.

3.2.2 *Dados Reais*

Realizado os testes com os dados gerados artificialmente, buscou-se efetuar outra simulação frente a um conjunto de dados proveniente de uma aplicação real. A saber, foram utilizados dados oriundos da análise para concessão de crédito[6] para clientes de instituições financeiras de Taiwan [UCI (2016)]. Os dados são formados por 24 atributos pertinentes à averiguação de concessão de crédito a um determinado cliente que é referenciado por cada instância de dados, o conjunto é composto por 30000 delas. Cada instância é classificada por 1 (crédito aprovado) ou -1 (crédito não aprovado).

Como já discutido, para esse caso foi necessário pré processar os dados para, por exemplo, tratar dos atributos faltantes através da imputação pela média. Pois dessa forma, a instancia não é inteiramente descartada e este atributo passa a não agregar informação à média e nem à variância.

De forma análoga a simulação com os dados artificiais, os parâmetros pertinentes tanto ao *SVM* distribuído quanto ao centralizado foram determinados a partir de uma busca em grade, a saber, os parâmetros testados são dados pela Tabela 4.

Tabela 4 – Parâmetros pertinentes a busca em grade à classificação linear com os dados reais.

SVM Distribuído		*SVM* Central
C	c	c
2^{-5}	1	2^{-5}
2^{-2}	10	2^{-2}
-	-	2
-	-	2^2
-	-	2^5

A partir do teste em `reallinear.py`, foi gerada uma rede conexa com 30 nós e os métodos distribuído e centralizado foram treinados frente ao conjunto de dados para análise de crédito. Ademais, utilizando o método em `plotrisk.py` foi gerado um gráfico (Figura 8) que apresenta o risco determinado pelo modelo distribuído a cada 5 iterações durante as primeiras 400 iterações. Para efeito de comparação, também são

[6] De fato, o método de máquina de suporte vetorial é largamente utilizado para a análise de crédito [Louzada, Ara e Fernandes (2016)].

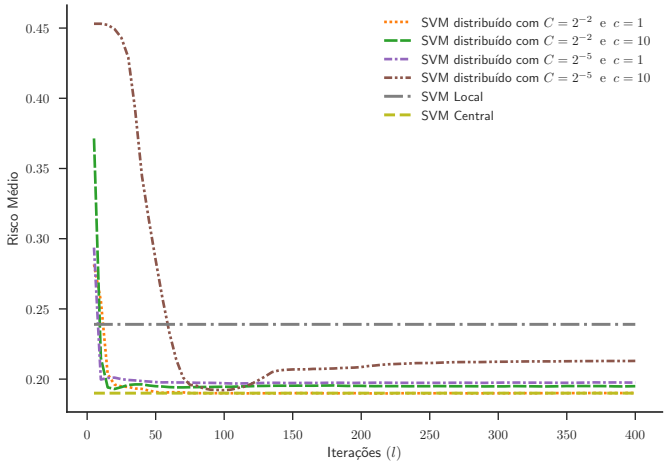


Figura 8 – Os riscos apresentados pelo modelo distribuído a cada 5 iterações entre as 400 primeiras iterações do Algoritmo 8, comparados aos riscos provenientes dos modelos central e local.

apresentados os riscos determinados pelo método centralizado aplicado ao conjunto total e local de dados após 400 iterações. Note que, neste caso a busca em grade realizada para o *SVM* distribuído é detalhada para cada conjunto de parâmetros C, c, o que difere do caso anterior com os dados artificiais em que foi tomado apenas os parâmetros ótimos entre os pertencentes a grade manualmente fixada.

Dessa forma, novamente, pode-se inferir que o Algoritmo 8 converge a um modelo preditivo com acurácia semelhante a proveniente do *SVM* central aplicado ao conjunto total de dados. Ainda, o *SVM* distribuído exibe uma acurácia maior que a abordagem local independente dos parâmetros C e c testados. Fato que por si só já justificaria o uso do Algoritmo 8.

3.3 ESTUDOS DE CASO À CLASSIFICAÇÃO NÃO LINEAR

Nesta seção será apresentada outras duas simulações numéricas, contudo voltadas ao Algoritmo 9. É importante ressaltar que estas simulações podem ser reproduzidas através do código hospedado no GitHub[7].

3.3.1 *Geração dos Dados Comuns a Rede*

Como apresentado no capítulo anterior, o Algoritmo 9 demanda a existência de um conjunto dados comum a todos os agentes da rede, o qual é formado por um conjunto finito de p vetores em $\mathbb{R}^n$ denotados por χ_i. Contudo, a aplicação de interesse impõe que os dados de treino não sejam comunicados entre os agentes. Logo,

7 Hospedado em https://github.com/caiodadauto/Distributed-SVM.

esses vetores precisam ser gerados de forma artificial se baseando em informações do conjunto total de dados de treino que não comprometem a privacidade dos mesmos.

A saber, considere os valores

$$x_t^{\min} = \min_{\substack{i=1,\cdots,N \\ j=1,\cdots,m_i}} \{[x_{ij}]_t\} \qquad x_t^{\max} = \max_{\substack{i=1,\cdots,N \\ j=1,\cdots,m_i}} \{[x_{ij}]_t\}$$

em que o índice t representa a t-ésima coordenada do vetor x_{ij}, o qual é o j-ésimo dado do nó i. Cada coordenada t de um dado vetor χ_i é definida por um valor aleatório proveniente de uma distribuição uniforme limitada por $(x_t^{\min}, x_t^{\max})$. Por fim, aplicando esse processo p vezes os vetores $\{\chi_i\}_{i=1,\cdot,p}$ comum a todos os agentes da rede são gerados.

3.3.2 *Dados Artificiais*

Foram gerados artificialmente 2560 dados bidimensionais de forma que a distribuição espacial de cada classe associada a esses dados simulasse uma malha de xadrez, como apresentado na Figura 9. Frente a esses dados, foram treinados três modelos preditivos não lineares provenientes do *SVM* distribuído não linear e *SVM* central não linear aplicado ao conjunto total e local de dados, é importante ressaltar que este último é feito sem cooperação dos demais agentes da rede. Ambos os métodos para o *SVM* não linear foram testados com o núcleo *RBF* (Tabela 1).

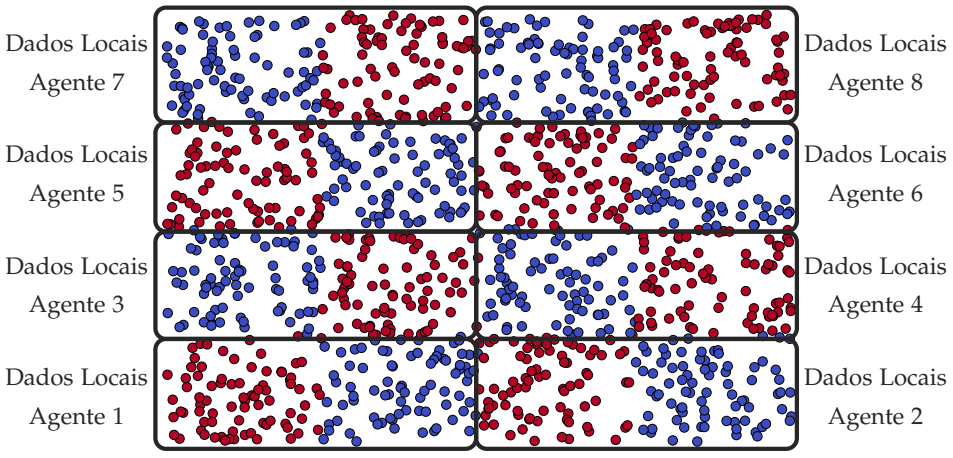


Figura 9 – Distribuição dos dados locais entre os agentes da rede para o conjunto de dados em malha de xadrez.

Esta simulação busca apenas ilustrar o impacto da não cooperação entre os agentes da rede e, principalmente, mostrar que o Algoritmo 9 retorna resultados satisfatórios mesmo quando os dados locais induzem a uma classificação que claramente não condiz com o comportamento global. Para isso, o conjunto de dados foi

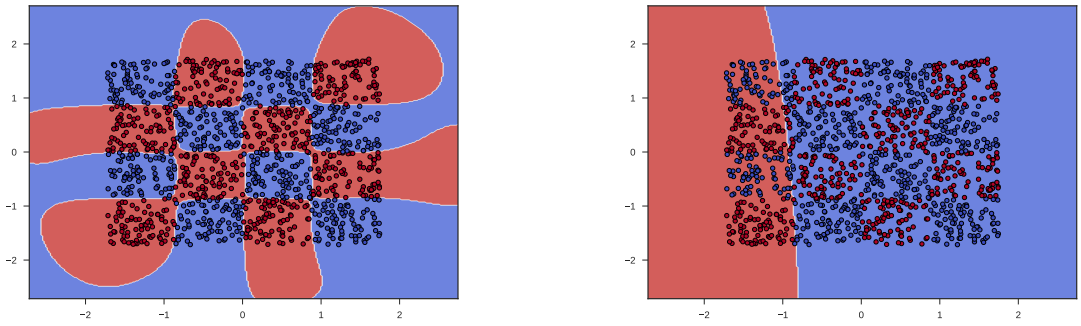
dividido de forma que cada agente possua exatamente um par de regiões adjacentes de classes alternadas que compõe a malha de xadrez, como pode ser visualizado na Figura 9. Ainda, note que, devido ao tamanho da malha de dados gerada, esses testes serão realizados necessariamente para uma rede com 8 nós.

Dessa forma, os três modelos preditivos foram treinados usando esse conjunto de dados com a distribuição local ilustrada na Figura 9. Os parâmetros de cada modelo foram determinados com a busca em grade definida pelos valores apresentados na Tabela 5. Enquanto, a dimensão p necessária ao Algoritmo 9 foi tomada como igual a 150 e os vetores χ_i foram gerados como descrito na subseção 3.3.1.

Tabela 5 – Parâmetros pertinentes a busca em grade à classificação não linear com os dados artificiais.

SVM Distribuído			*SVM* Central	
C	c	γ	C	γ
1	1	2^{-3}	1	2^{-5}
2^5	2	2^{-2}	2	2^{-4}
2^6	8	2^{-1}	2^2	2^{-3}
-	-	1	2^3	2^{-2}
-	-	2	2^4	2^{-1}
-	-	-	2^5	1
-	-	-	2^6	2

Determinado os parâmetros ótimos contidos nessa grade, os modelos distribuído e centralizado treinados com esse conjunto de dados podem, utilizando o método em `plotnonlinearclassifier.py`, ser visualizados na Figura 10 e Figura 11. Observe que o classificador local não condiz com o comportamento global dos dados,

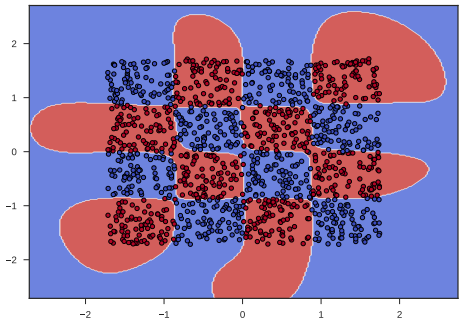


(a) Classificador oriundo do *SVM* central aplicado ao conjunto total de dados.

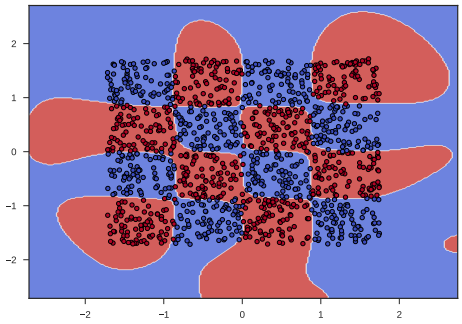
(b) Classificador oriundo do *SVM* central aplicado ao conjunto local do agente 1.

Figura 10 – Classificadores não lineares provenientes do *SVM* central.

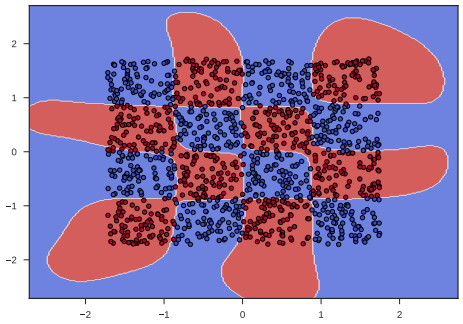
pois o mesmo procura determinar um modelo à classificação dos seus dados, o que



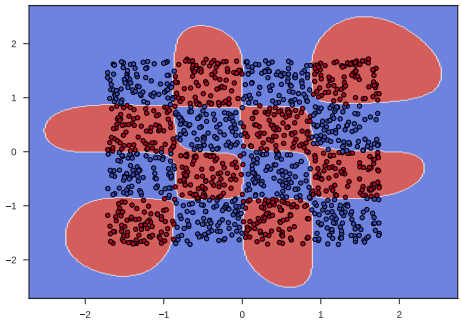
(a) Agente 1



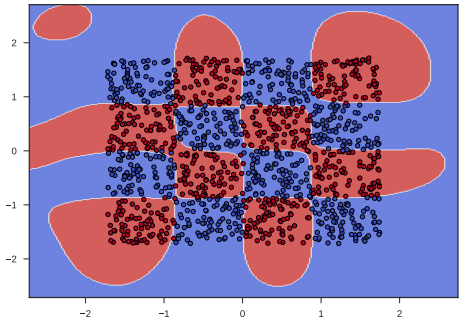
(b) Agente 2



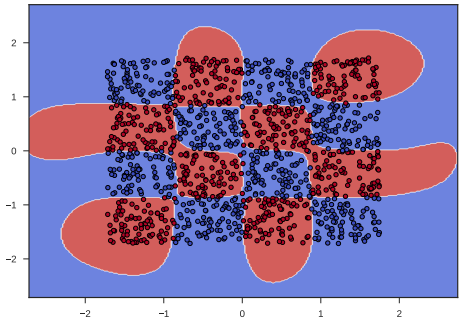
(c) Agente 3



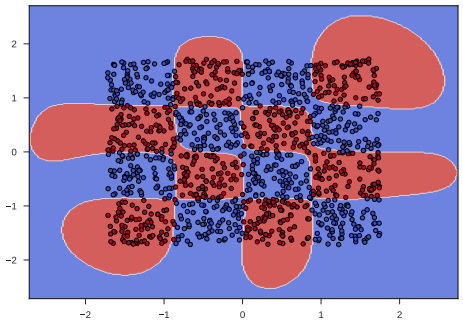
(d) Agente 4



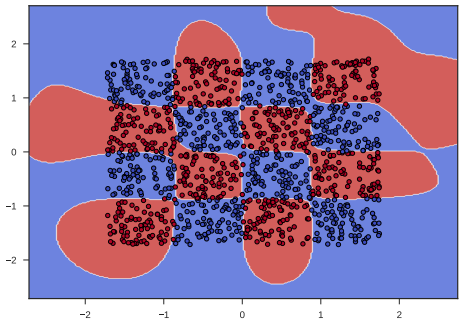
(e) Agente 5



(f) Agente 6



(g) Agente 7



(h) Agente 8

Figura 11 – Classificadores não lineares para cada agente da rede gerados pelo Algoritmo 9.

claramente não satisfaz a distribuição imposta pela malha de xadrez.

Por outro lado, a Figura 11 apresenta os classificadores determinados por cada nó. note que estes não são iguais, como de fato não deveriam ser, uma vez que o Algoritmo 9 impõe apenas o consenso segundo a transformação $\Phi_\chi \omega_i$. Ademais, os modelos apresentado na Figura 11 refletem a característica não linear da malha de xadrez, mesmo com a distribuição dos dados locais imposta como apresentada na Figura 9.

3.3.3 *Dados Reais*

De forma análoga ao caso linear, é realizada outra simulação com um conjunto de dados proveniente de uma aplicação real. A saber, são utilizados dados oriundos do registro de 768 mulheres com no mínimo 21 anos que são diagnosticadas ou não com diabetes[8]. Esse diagnóstico foi realizado entre a população indígena Pima que vive próxima a Phoenix, Arizona, USA [UCI (1990)]. O conjunto de dados é composto por 8 atributos pertinentes ao diagnostico da diabetes e os rótulos utilizados para caracterizar a presença ou não da doença são 1 (com diabetes) e -1 (sem diabetes).

Da mesma forma que para a simulação no caso linear, ressalta-se que os atributos faltantes foram tratados através da imputação pela média.

Os parâmetros ideais para o *SVM* não linear distribuído e o centralizado foram determinados a partir de uma busca em grade, a saber, os parâmetros testados são dados pela Tabela 6. Quanto a dimensão da imagem de Φ_χ, foi utilizado p igual a 300.

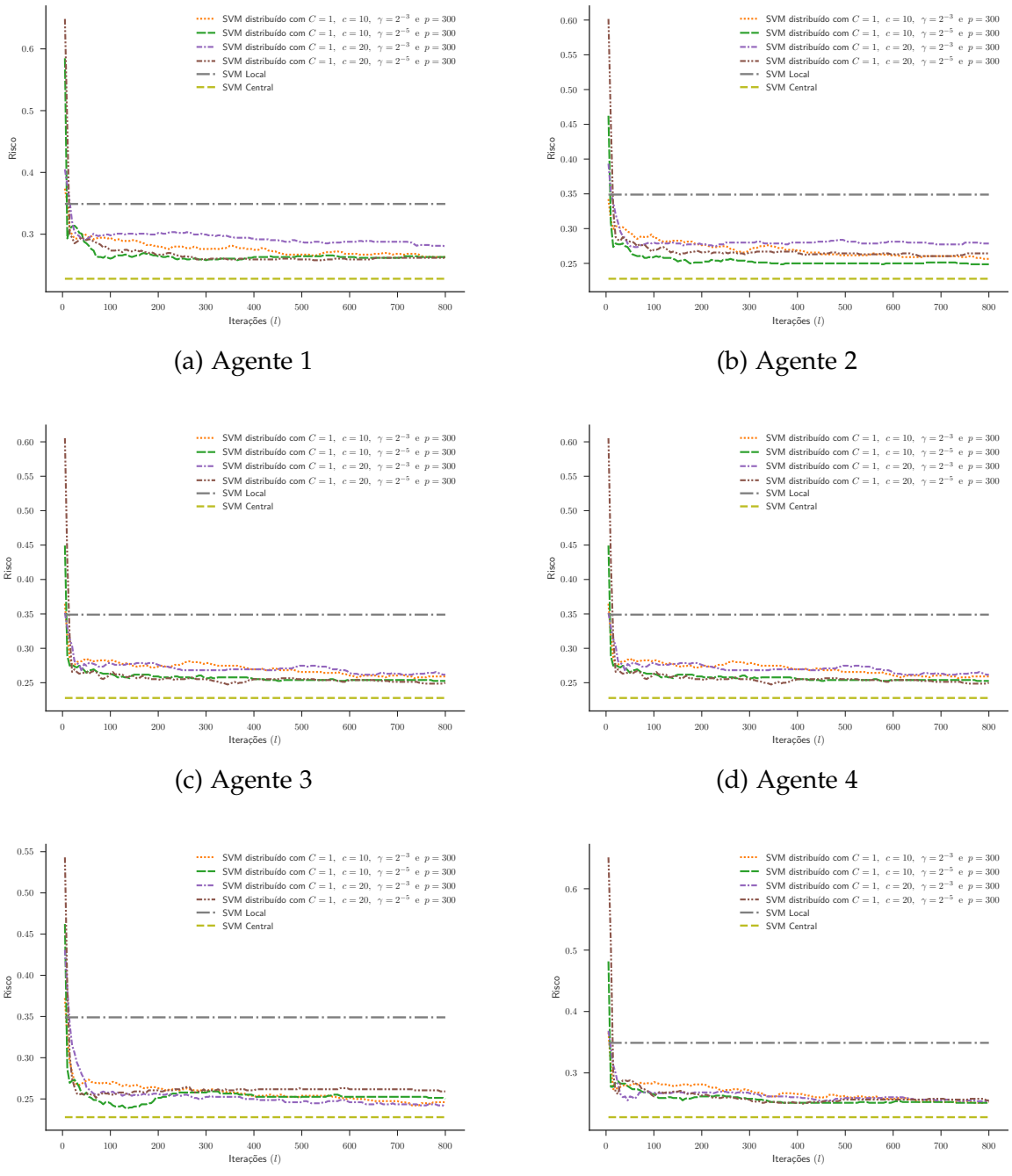
Tabela 6 – Parâmetros pertinentes a busca em grade à classificação não linear com os dados reais.

SVM Distribuído			*SVM* Central	
C	c	γ	C	γ
1	10	2^{-5}	1	2^{-8}
-	20	2^{-3}	2^2	2^{-5}
-	-	-	2^4	1
-	-	-	2^5	-
-	-	-	2^6	-

A partir do teste em `realnonlinear.py`, foi gerada uma rede conexa com 6 nós e os métodos distribuído e centralizado foram treinados usando o conjunto de

[8] O diagnóstico e interpretação dos dados de diabetes é, de fato, um importante problema de classificação [Barakat, Bradley e Barakat (2010)].

dados para o diagnóstico de diabetes. Ademais, utilizando o método em `plotrisk.py` foram gerados gráficos (Figura 12), em que cada um destes apresenta o risco determinado pelo modelo referente a um dado agente a cada 5 iterações durante as primeiras 800 iterações. Para efeito de comparação, também são apresentados os riscos



(a) Agente 1

(b) Agente 2

(c) Agente 3

(d) Agente 4

(e) Agente 5

(f) Agente 6

Figura 12 – Os riscos provenientes dos modelos originários de cada nó da rede determinados a cada 5 iterações entre as 800 primeiras iterações do Algoritmo 9, comparados com os riscos apresentados pelos modelos central e local.

determinados pelo método centralizado aplicado somente ao conjunto total e local de dados após 800 iterações. Dessa forma, pode-se inferir que o Algoritmo 9 converge a

um modelo preditivo com acurácia superior a proveniente do *SVM* central aplicado ao conjunto local de dados, fato que justifica a cooperação entre os agentes da rede.

CAPÍTULO 4

Considerações Finais

4.1 Trabalhos futuros

Há ainda, diversos aspectos a serem explorados no contexto de aprendizado supervisionado distribuído e descentralizado. A seguir são apresentados breves comentários acerca de alguns tópicos relevantes a esse contexto que não foram explorados nesta dissertação.

4.1.1 *Classificação Multi Classes*

Apesar dos métodos para o *SVM* distribuído discutidos nesta dissertação serem voltados apenas à classificação binária, é possível aplicar esses métodos à problemas de aprendizado supervisionado com conjunto de treino que apresenta mais de duas classes, problemas que são denominados por classificação multi classes. De fato, basta aplicar uma das 3 deferentes abordagens para a classificação multi classes comumente utilizadas [Hsu e Lin (2002)], denominadas por *one-against-all*, *one-against-one* e *DAGSVM*.

Basicamente, a abordagem *one-against-all* cria d modelos *SVM*, em que d é o número de classes que o conjunto de treino apresenta. O i-ésimo modelo é treinado utilizando as instâncias originalmente rotuladas pela classe i por 1 e as demais instâncias por -1, ou seja, o i-ésimo modelo é derivado do seguinte problema

$$\begin{aligned} \min \quad & \frac{1}{2}||w_i||^2 + C\sum_{l=1}^{m}\varepsilon_{il} \\ \text{s.a.} \quad & 1 - w_i^T x_l + b_i - \varepsilon_{il} \leqslant 0 \quad \text{se } y_l = i \\ & 1 + w_i^T x_l + b_i - \varepsilon_{il} \leqslant 0 \quad \text{se } y_l \neq i \\ & -\varepsilon_{il} \leqslant 0. \end{aligned} \tag{4.1}$$

Dessa forma, a classificação de um dado x é feita a partir da seguinte expressão

$$\arg\min_{i=1,\cdots,d} \left\{ w_i^T x + b_i \right\}. \tag{4.2}$$

Por outro lado, a abordagem *one-against-one* cria $\frac{d(d-1)}{2}$ modelos, em que cada um desses é derivado do treino utilizando apenas as instâncias que possuem um dado par de classes, a saber i e j. Ou seja,

$$\begin{aligned} \min \quad & \frac{1}{2}||w_{ij}||^2 + C\sum_{l=1}^{m}\varepsilon_{ijl} \\ \text{s.a.} \quad & 1 - w_{ij}^T x_l + b_{ij} - \varepsilon_{ijl} \leqslant 0 \quad \text{se } y_l = i \\ & 1 + w_{ij}^T x_l + b_{ij} - \varepsilon_{ijl} \leqslant 0 \quad \text{se } y_l = j \\ & -\varepsilon_{ijl} \leqslant 0. \end{aligned} \tag{4.3}$$

Neste caso, há duas maneiras [Hsu e Lin (2002)] de se classificar um dado x. Na primeira a classe de x é determinada pela votação entre os modelos. Já na segunda, a classe é determinada por uma arvore de decisão que utiliza os modelos em cada nó. Esta última alternativa é a que define a abordagem *DAGSVM*.

Dessa forma, observe que no caso distribuído basta aplicar um desses processos a cada nó, em que o algoritmo utilizado depende da natureza linear ou não do problema.

4.1.2 *Regressão*

Além dos problemas de classificação, é recorrente entre problemas de aprendizado supervisionado o uso de conjuntos de treino em que cada instância é associada a um número real. Neste caso, o problema é caracterizado como um problema de regressão, em que não há classes a serem determinadas, mas se determina a função que melhor descreve o comportamento dos dados de treino.

O *SVM* pode ser modificado de forma a se enquadrar a classe de problemas de regressão supervisionada, como descrito em Smola e Schölkopf (2004). Então, o *SVM* pode ser reformulado como segue

$$\begin{aligned} \min \quad & \frac{1}{2}||w||^2 + C\sum_{j=0}^{m}(\zeta_j + \tilde{\zeta}_j) \\ & y_j - (w^T x_j + b) - (\epsilon + \zeta_j) \leqslant 0 \\ & w^T x_j + b - y_j - (\epsilon + \tilde{\zeta}_j) \leqslant 0 \\ & -\zeta_j, -\tilde{\zeta}_j \leqslant 0, \end{aligned} \tag{4.4}$$

em que ϵ é um novo parâmetro que dita a precisão com o problema aproxima a função aos dados de treino.

Note que a formulação (4.4) se assemelha ao *SVM* aplicado a classificação supervisionada. Desse fato, espera-se que a manipulação teórica para o *SVM* aplicado a regressão supervisionada distribuída siga o mesmo raciocínio apresentado nesta dissertação para a classificação supervisionada distribuída.

4.1.3 *SVM Online*

Outra situação importante ocorre quando o conjunto de treino é modificado durante a execução do método, por exemplo, nos casos de *data stream*. Essa situação é, também, abordada em Forero, Cano e Giannakis (2010), porém sem nenhum tratamento teórico. A saber, é sugerido que as matrizes X_i e Y_i definidas durante a formulação distribuída sejam modificadas, quando necessário, a cada iteração do método. Por exemplo, o Algoritmo 8 passaria ser dado por

Algoritmo 10 – LDSVM Online

Dados: $\lambda_i^0 = 0$ e v_i^0 para todo $i = 1, \cdots, N$
para cada *nó i* **faça**
  para $l = 0$, 1, 2, 3, ... **faça**

$$\begin{aligned}
\mu_i^{l+1} &\in \arg\max_{0 \leqslant \mu_i \leqslant C1_i} \left\{ -\frac{1}{2}\mu_i^T Y_i^l X_i^l D_i^{-1} (X_i^l)^T Y_i^l \mu_i + (1_i + Y_i^l X_i^l D_i^{-1} r_i^l)^T \mu_i \right\} \\
v_i^{l+1} &= D_i^{-1} \left[(X_i^l)^T Y_i^l \mu_i^{l+1} - r_i^l \right] \\
\lambda_i^{l+1} &= \lambda_i^l + \frac{c}{2} \sum_{k \in \mathbb{K}_i} (v_i^{l+1} - v_k^{l+1})
\end{aligned}$$

  com

$$\begin{aligned}
r_i^l &= 2\lambda_i^l - c \sum_{k \in \mathbb{K}_i} (v_i^l + v_k^l) \\
D_i &= (I - e_{n+1} e_{n+1}^T) + 2c\#\mathbb{K}_i I
\end{aligned}$$

  fim
fim

4.1.4 *Critério de Convergência*

Nesta dissertação não foi utilizado um critério de convergência, contudo é natural que tal critério seja conveniente às aplicações de interesse. Na seção 3.3 de Boyd et al. (2010) é apresentado uma alternativa para o critério de convergência do método *ADMM*. O qual deriva das condições de otimalidade da classe de problemas de interesse ao *ADMM* (1.37). O critério de convergência é dado por

$$f(x^l) + g(x^l) - (f^* + g^*) \leqslant ||\lambda^l||\,||r^l|| + \alpha||s^l|| < \varepsilon, \tag{4.5}$$

em que $\varepsilon > 0$ define a precisão desejada, α é uma estimativa tal que $||x^l - x^*|| \leqslant \alpha$, $r^l = Ax^l + Bz^l - d$ e $s^l = cA^T B(z^{l+1} - z^l)$.

Por outro lado, no contexto do *SVM* linear distribuído, aplica-se esse critério de convergência ao problema (2.44) de forma que

$$F(v^l) - (F^*) \leqslant ||\lambda^l||||r^l|| + \alpha||s^l|| < \varepsilon \tag{4.6}$$

com $r^l = Av^l + Bu^l$ e $s^l = cA^TB(u^{l+1} - u^l)$. Dessa forma, utilizando as definições de v, u, λ, A e B introduzidas para esse problema, as variáveis r^l e s^l podem ser determinadas de forma separável entre os nós da rede e, assim, computadas de forma distribuída. Nesse caso, intuitivamente, para satisfazer o critério de convergência será necessária a comunicação entre todos os nós das parcelas que compõe a soma $||\lambda^l||||r^l|| + \alpha||s^l||$, fato que pode levar a um demasiado tráfego de informação na rede. Caso isso ocorra, uma solução imediata seria averiguar o critério de convergência a cada $k > 2$ iterações, ao invés de averiguá-lo em toda iteração do método.

4.2 CONCLUSÃO

Nesta dissertação foram apresentadas duas variantes para o problema do *SVM* aplicado a uma rede descentralizada com conjunto de treino distribuído: a classificação linear (Algoritmo 8) e não linear (Algoritmo 9). Ambas, foram derivadas através da aplicação do método *ADMM*. A partir da convergência desse método foi possível garantir a convergência das duas variantes para o *SVM* distribuído, em particular a classificação linear converge ao *SVM* aplicado ao conjunto de treino centralizado.

Além disso, foram realizadas simulações numéricas que evidenciam a vantagem do uso do método proposto em detrimento da aplicação direta do *SVM* ao conjunto local de dados sem a cooperação com os demais agentes da rede. Por fim, foram apresentadas rapidamente variações do uso do *SVM* aplicado ao aprendizado supervisionado distribuído descentralizado para lidar com a classificação multi classes, a regressão e aplicações que exigem a modificação contínua dos dos dados de treino.

Referências

AGGARWAL, C. C.; HINNEBURG, A.; KEIM, D. A. On the surprising behavior of distance metrics in high dimensional spaces. 2001. Citado na página 50.

BARAKAT, N.; BRADLEY, A. P.; BARAKAT, M. N. H. Intelligible support vector machines for diagnosis of diabetes mellitus. *IEEE Transactions on Information Technology in Biomedicine*, v. 14, n. 4, p. 1114–1120, 2010. Citado na página 90.

BELLMAN, R. E. *Dynamic Programming*. 1. ed. [S.l.]: Princeton University Press, 1957. Citado na página 50.

BERLINET, A.; THOMAS-AGNAN, C. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. 1. ed. [S.l.]: Springer, 2004. Citado 2 vezes nas páginas 101 e 102.

BERTSEKAS, D. P. *Convex Optimization Theory*. [S.l.]: Athena Scientific, 2009. Citado 4 vezes nas páginas 22, 25, 29 e 35.

_____. *Convex Optimization Algorithms*. 1. ed. [S.l.]: Athena Scientific, 2015. Citado 4 vezes nas páginas 28, 29, 30 e 34.

_____. *Nonlinear Programming*. 3. ed. [S.l.]: Athena Scientific, 2016. Citado 2 vezes nas páginas 22 e 23.

BERTSEKAS, D. P.; TSITSIKHS, J. N. *Parallel and Distributed Computation: Numerical Methods*. 2. ed. [S.l.]: Athena Scientific, 1997. Citado na página 38.

BIRGIN, E. G.; MARTíNEZ, J. M. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, n. 22, p. 101–125, 2002. Citado na página 64.

BONDY, J. A.; MURTY, U. S. R. *Graph Theory*. 1. ed. [S.l.]: Springer, 2008. Citado na página 53.

BOYD, S.; PARIKH, N.; CHU, E.; PELEATO, B.; ECKSTEIN, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, n. 1, p. 1–122, 2010. Citado 2 vezes nas páginas 18 e 95.

CHAPELLE, O.; VAPNIK, V.; BOUSQUET, O.; MUKHERJEE, S. Choosing multiple parameters for support vector machines. p. 131, 2002. Citado na página 79.

CVXOPT. versão 1.1.9. Disponível em: <http://cvxopt.org/>. Citado na página 79.

FORERO, P. A.; CANO, A.; GIANNAKIS, G. B. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, n. 11, p. 1663–1707, 2010. Citado 4 vezes nas páginas 18, 51, 65 e 95.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. [S.l.]: Springer, 2013. Citado 2 vezes nas páginas 47 e 48.

HSU, C.-W.; LIN, C.-J. A comparison of methods for multiclass support vector machines. *Trans. Neur. Netw.*, IEEE Press, v. 13, n. 2, 2002. Citado 2 vezes nas páginas 93 e 94.

HSU, C. wei; CHANG, C. chung; LIN, C. jen. A practical guide to support vector classification. 2010. Citado 2 vezes nas páginas 78 e 79.

JERZAK, Z.; ZIEKOW, H. The debs 2014 grand challenge. *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, p. 266–269, 2014. Citado na página 17.

KLATZER, T.; POCK, T. Continuous hyper-parameter learning for support vector machines. *Computer Vision Winter Workshop*, n. 20, 2015. Citado na página 79.

KREYSZIG, E. *Introductory Functional Analysis with Applications*. 4. ed. [S.l.]: John Wiley & Sons, 2016. Citado 4 vezes nas páginas 101, 102, 103 e 105.

LOUZADA, F.; ARA, A.; FERNANDES, G. B. Classification methods applied to credit scoring: Systematic review and overall comparison. 2016. Citado 2 vezes nas páginas 18 e 85.

LUENBERGER, D. G.; YE, Y. *Linear and Nonlinear Programming*. 4. ed. [S.l.]: Springer, 2016. Citado 3 vezes nas páginas 22, 49 e 62.

MATPLOTLIB. versão 2.0.1. Disponível em: <http://matplotlib.org/>. Citado na página 78.

MEYER, C. D. *Matrix Analysis and Applied Linear Algebra*. 2. ed. [S.l.]: SIAM, 2000. Citado na página 73.

MPI4PY. versão 2.0.0. Disponível em: <https://pypi.python.org/pypi/mpi4py>. Citado na página 79.

NUMPY. versão 1.13.0. Disponível em: <http://www.numpy.org/>. Citado na página 78.

PANDAS. versão 0.20.2. Disponível em: <http://pandas.pydata.org/>. Citado na página 78.

PLATT, J. C. Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report Microsoft Research*, 1998. Citado na página 49.

SANFORD, J. F.; POTKONJAK, M.; SLIJEPCEVIC, S. Localization in wireless networks. p. 41–64, 01 2012. Citado na página 18.

SCARDAPANE, S.; ROSA, A.; CICCARELLI, V.; UNCINI, A.; PANELLA, M. Privacy-preserving data mining for distributed medical scenarios. p. 119–128, 2018. Citado na página 17.

SCHöLKOPF, B.; SMOLA, A. *Learning with Kernels*. [S.l.]: MIT Press, 2002. Citado na página 106.

SCIKIT-LEARN. versão 0.18. Disponível em: <http://scikit-learn.org/stable/index.html>. Citado 3 vezes nas páginas 78, 80 e 82.

SEABORN. versão 0.7.1. Disponível em: <https://seaborn.pydata.org/>. Citado na página 78.

SHIERS, J. The worldwide lhc computing grid (worldwide lcg). *Computer Physics Communications*, v. 177, p. 219–223, 07 2007. Citado na página 17.

SMOLA, A. J.; SCHöLKOPF, B. A tutorial on support vector regression. 2004. Citado na página 94.

UCI. *Pima Indians Diabetes*. 1990. Disponível em: <https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes>. Citado na página 90.

_____. *Default of Credit Card Clients Data Set*. 2016. Disponível em: <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>. Citado na página 85.

Apêndices

APÊNDICE A

Caracterização de Núcleos

Neste apêndice, procura-se introduzir o ferramental necessário para a compreensão das manipulações realizadas durante a abordagem não linear do problema de suporte vetorial tanto centralizado quanto distribuído. A saber, serão apresentados alguns resultados relevantes com relação à espaços de *Hilbert* reproduzidos por núcleos[1] se limitando a espaços reais.

A.1 Definições e resultados preliminares

Nessa seção será apresentado aspectos teóricos básicos de análise funcional que podem ser encontrados com mais detalhes em Kreyszig (2016) e Berlinet e Thomas-Agnan (2004).

Definição 6 (Espaço de Hilbert). *Um espaço vetorial $\mathcal{H}$ é dito de Hilbert se é dotado de produto interno, isto é, uma função $\langle \cdot, \cdot \rangle : \mathcal{H} \mapsto \mathbb{R}$ que satisfaz as seguintes propriedades para todo $x, y, z \in \mathcal{H}$:*

(i) $\langle x, y \rangle = \langle y, x \rangle$

(ii) $\langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$ *com* $\alpha, \beta \in \mathbb{R}$

(iii) $\langle x, x \rangle \geqslant 0$ *e* $\langle x, x \rangle = 0$ *se somente se* $x = 0$

Ademais, $\mathcal{H}$ é dotado de norma induzida pelo produto interno, ou seja, $||x|| = \sqrt{\langle x, x \rangle}$ e é um espaço completo segundo a métrica induzida por esta norma.

Definição 7 (Funcional Linear). *Uma função f sobre um espaço vetorial V é dita funcional linear se se é linear em V e $f : V \mapsto \mathbb{R}$.*

[1] Comumente denominado como *Reproducing Kernel Hilbert Spaces (RKHS)*.

Definição 8 (Função Avaliação). *Seja $\mathcal{F}$ um espaço de Hilbert composto por funcionais lineares $f : V \mapsto \mathbb{R}$. A função $L_x : \mathcal{F} \mapsto \mathbb{R}$ é dita função avaliação sobre $\mathcal{F}$ se, para todo vetor $x \in V$, tem-se que*

$$L_x(f) = f(x) \quad \forall f \in \mathcal{F}.$$

Observe que L_x é, também, um funcional linear.

Definição 9 (Função Positiva). *Uma função $K : V \times V \mapsto \mathbb{R}$ é dita positiva se a matriz*

$$[K(x_i, x_j)]_{1 \leqslant i,j \leqslant n} = \begin{bmatrix} K(x_1,x_1) & \cdots & K(x_n,x_1) \\ \vdots & \cdots & \vdots \\ K(x_n,x_1) & \cdots & K(x_n,x_n) \end{bmatrix}$$

for semi-positiva definida para qualquer $n \in \mathbb{N} - \{0\}$ e quaisquer $x_1, \cdots, x_n \in V$.

Ademais, considere o teorema da representação de *Riesz* derivado na seção 3.8 de Kreyszig (2016) e enunciado na Teorema 2.

Teorema 2 (Teorema da Representação de Riesz). *Seja um funcional linear $f : \mathcal{H} \mapsto \mathbb{R}$ sobre um espaço de Hilbert $\mathcal{H}$, em que f é limitado, ou seja, existe $M > 0$ tal que*

$$|f(x)| \leqslant M||x||_{\mathcal{H}} \qquad \forall x \in \mathcal{H}.$$

Então, f pode ser representado por um produto interno, a saber existe $z \in \mathcal{H}$ tal que

$$f(x) = \langle x, z \rangle_{\mathcal{H}},$$

em que z depende de f e é unicamente determinado por este funcional. Em particular,

$$||z||_{\mathcal{H}} = ||f||_{\mathcal{F}},$$

em que a norma de f é a norma induzida por

$$||f||_{\mathcal{F}} = \sup_{\substack{x \in \mathcal{H} \\ ||x||_{\mathcal{H}} = 1}} |f(x)|.$$

Note que segue diretamente desse teorema o fato de que $\mathcal{F}$ é um espaço de *Hilbert*.

A.2 ESPAÇOS DE HILBERT REPRODUZIDOS POR NÚCLEOS

Nesta seção são apresentados resultados pertinentes à caracterização dos núcleos. Resultados, estes, que podem ser encontrados em Berlinet e Thomas-Agnan (2004).

Definição 10 (RHS). *Um espaço de Hilbert $\mathcal{F}$ formado por funcionais lineares $f : V \mapsto \mathbb{R}$ é denominado como espaço de Hilbert ou RHS se a função avaliação L_x sobre $\mathcal{F}$ é contínua.*

Note que, neste caso, como L_x é contínuo, então é, também, limitado[2], ou seja,

$$|L_x(f)| = |f(x)| \leqslant M||f||_{\mathcal{F}}$$

Ainda, como L_x é limitado, então vale a seguinte proposição.

Proposição 10. *Seja $\{f_n\}$ uma sequência de funções em um RKHS $\mathcal{F}$ que converge a uma função f desse espaço. Então, $\{f_n\}$ converge pontualmente a f.*

Demonstração. Considere $f_n \to f$ e um $x \in V$ qualquer. Então, para qualquer $\epsilon > 0$ existe $N > 0$ tal que

$$|f_n(x) - f(x)| = |L_x(f_n) - L_x(f)| = |L_x(f_n - f)| \leqslant M||f_n - f||_{\mathcal{F}} < M\epsilon \quad \forall n \geqslant N,$$

ou seja, f_n converge pontualmente à f.

■

Por outro lado, define-se o que vem a ser um núcleo, a saber

Definição 11 (Núcleo). *Seja um espaço de Hilbert $\mathcal{F}$ formado por funcionais lineares $f : V \mapsto K$. Uma função $K : V \times V \mapsto \mathbb{R}$ é dita núcleo associado a $\mathcal{F}$ se satisfazer para todo $x \in V$:*

(i) $K(\cdot, x) \in \mathcal{F}$

(ii) $\langle f, K(\cdot, x)\rangle_{\mathcal{F}} = f(x)$

Observe que, diretamente da condição *(ii)*, é possível derivar que o núcleo K é simétrico e positivo. Com efeito,

$$\langle K(\cdot, y), K(\cdot, x)\rangle_{\mathcal{F}} = K(x, y)$$

é trivialmente simétrico devido a definição de produto interno. Enquanto, tomando $n > 0$ natural, $a_1, \cdots, a_n \in \mathbb{R}$ e $x_1, \cdots, x_n \in V$ quaisquer, tem-se que

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K(x_i, x_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \langle K(\cdot, x_i), K(\cdot, x_j)\rangle_{\mathcal{F}} = \left\|\sum_{i=1}^{n} a_i K(\cdot, x_i)\right\|_{\mathcal{F}}^2 \geqslant 0.$$

Logo, K é uma função positiva.

Proposição 11 (Unicidade do Núcleo). *Seja $\mathcal{F}$ um espaço de Hilbert. Se houver um núcleo K associado a esse espaço, esse núcleo é único.*

[2] Como derivado no teorema 2.8-3 de Kreyszig (2016).

Demonstração. Suponha que $\mathcal{F}$ possua dois núcleos associados, a saber K_1 e K_2. Assim, para todo $x \in V$ e $f \in \mathcal{F}$, tem-se que

$$\begin{aligned} \langle f, K_1(\cdot,x) - K_2(\cdot,x)\rangle_{\mathcal{F}} &= \langle f, K_1(\cdot,x)\rangle_{\mathcal{F}} - \langle f, K_2(\cdot,x)\rangle_{\mathcal{F}} \\ &= f(x) - f(x) \\ &= 0 \end{aligned}$$

Tomando, $f = K_1(\cdot,x) - K_2(\cdot,x)$, conclui-se que

$$||K_1(\cdot,x) - K_2(\cdot,x)||^2_{\mathcal{F}} = 0 \quad \Leftrightarrow \quad K_1(\cdot,x) = K_2(\cdot,x).$$

■

Agora, o seguinte teorema relaciona o conceito de espaço de *Hilbert* de reprodução com o conceito de núcleo.

Proposição 12. *Um espaço $\mathcal{F}$ é RHS se somente se $\mathcal{F}$ possui um núcleo associado.*

Demonstração. Seja $\mathcal{F}$ *RHS*, então a função avaliação L_x sobre $\mathcal{F}$ é limitada. Assim, a partir do teorema da representação de *Riesz*, existe $k_x \in \mathcal{F}$ único para L_x tal que

$$L_x(f) = \langle f, k_x\rangle_{\mathcal{F}}.$$

Definindo $K(x,y) = k_x(y)$, segue que $k_x = K(\cdot,x)$. Dessa forma, $K(\cdot,x)$ caracteriza o núcleo sobre $\mathcal{F}$. De fato, claramente $K(\cdot,x) \in \mathcal{F}$ e

$$L_x(f) = \langle f, K(\cdot,x)\rangle = f(x).$$

Por outro lado, seja um espaço $\mathcal{F}$ de *Hilbert* que possua o núcleo K associado. Então, a partir da desigualdade de *Cauchy-Schwarz*, é válida a seguinte expressão para a função avaliação sobre $\mathcal{F}$,

$$|L_x(f)| = |f(x)| = |\langle f, K(\cdot,x)\rangle_{\mathcal{F}}| \leqslant ||f||_{\mathcal{F}}||K(\cdot,x)||_{\mathcal{F}} = \sqrt{K(x,x)}||f||_{\mathcal{F}}.$$

Logo, a função avaliação é limitada e, portanto, $\mathcal{F}$ é *RHS*.

■

Desa forma, espaços de *Hilbert* de reprodução são ditos reproduzidos por núcleo ou, comumente denominado *RKHS*.

Teorema 3 (Moore-Aronszajn)**.** *Seja $K : V \times V \mapsto \mathbb{R}$ uma função simétrica e positiva. Existe apenas um único espaço de Hilbert $\mathcal{F}$ formado por funcionais lineares $f : V \mapsto \mathbb{R}$ que é reproduzido por essa função K, ou seja, existe $\mathcal{F}$ único tal que K é o núcleo associado a $\mathcal{F}$. Ademais, $\mathcal{F}$ é separável.*

Demonstração. Seja K uma função simétrica e positiva, considere o seguinte espaço vetorial infinitamente gerado,

$$\mathcal{F}_0 = \text{span}\{K(\cdot,x) \mid x \in V\} = \left\{ f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot,x_i) \mid n \in \mathbb{N},\ x_i \in V \text{ e } \alpha_i \in \mathbb{R} \right\},$$

o qual é dotado o produto interno $\langle \cdot,\cdot \rangle$ definido por

$$\langle f,g \rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \beta_i K(x_i,x_j) \quad \forall g,f \in \mathcal{F}_0,$$

em que $f = \sum_{i=1}^{n} \alpha_i K(\cdot,x_i)$ e $g = \sum_{j=1}^{m} \beta_j K(\cdot,x_j)$. Dessa forma, note que o produto interno pode ser reescrito apenas em função de g ou de f. De fato,

$$\langle f,g \rangle = \sum_{i=1}^{n} \alpha_i g(x_i) = \sum_{j=1}^{m} \beta_j f(x_j) \quad \forall g,f \in \mathcal{F}_0.$$

Tomando o caso particular para $g = K(\cdot,x)$, tem-se que

$$\langle f,K(\cdot,x) \rangle = \sum_{i=1}^{n} \alpha_i K(x,x_i) = f(x) \quad \forall f \in \mathcal{F}_0.$$

Ou seja, K é núcleo associado a $\mathcal{F}_0$. Logo, de forma análoga a demonstração do Proposição 12, conclui-se que a função avaliação em $\mathcal{F}_0$ é limitada e, portanto, contínua.

Agora, considere o completamento de $\mathcal{F}_0$, denotado por $\mathcal{F}$. Seja uma sequência $\{f_n\}$ em $\mathcal{F}_0$ com limite $f \in \mathcal{F}$, então $\{f_n\}$ converge pontualmente a f. Assim, pela continuidade de $\langle \cdot,\cdot \rangle$ e para todo $x \in V$ fixo, tem-se que

$$f(x) = \lim_{n\to\infty} f_n(x) = \lim_{n\to\infty} \langle f_n,K(\cdot,x) \rangle = \langle f,K(\cdot,x) \rangle \quad \forall f \in \mathcal{F}_0$$

Portanto, K é núcleo associado a $\mathcal{F}$, o qual é *RKHS* e, ainda, é separável. De fato, $\mathcal{F}_0$ é contável e denso em $\mathcal{F}$.

Por contradição, considere que exista outro espaço associado a K, a saber $\mathcal{W}$. Como $\mathcal{W}$ é *RKHS* associado a K, então $K(\cdot,x) \in \mathcal{W}$ para todo $x \in V$, ou seja, $\mathcal{F}, \subset \mathcal{W}$. Ainda, como $\mathcal{W}$ é completo $\mathcal{F} \subset \mathcal{W}$. Agora, note que, como $\mathcal{F}$ é um subespaço fechado de $\mathcal{W}$, $W = \mathcal{F} \oplus \mathcal{F}^{\perp}$ [3]. Dessa forma, qualquer $f \in \mathcal{W}$ pode ser escrito como $f = g + g^{\perp}$, em que $g \in \mathcal{F}$ e $g^{\perp} \in \mathcal{F}^{\perp}$. Logo, utilizando o fato de K ser núcleo associado a $\mathcal{W}$ e $K(\cdot,x) \in \mathcal{F}$, conclui-se que

$$f(x) = \langle f,K(\cdot,x) \rangle = \left\langle g + g^{\perp},K(\cdot,x) \right\rangle = \langle g,K(\cdot,x) \rangle = g(x)$$

Assim, qualquer elemento de $\mathcal{W}$ pertence a $\mathcal{F}$, ou seja, $\mathcal{F} = \mathcal{W}$. Portanto, $\mathcal{F}$ é único.

■

[3] Resultado apresentado no teorema 3.3-4 de Kreyszig (2016).

Dessa forma, fica claro que basta determinar um núcleo simétrico e positivo para determinar um espaço de *Hilbert* reproduzível por esse núcleo, em que este espaço e núcleo são únicos. Contudo, não é possível representar vetores infinitamente gerados em uma unidade de processamento. Logo, no contexto prático de aprendizado de máquina, tentar solucionar um problema de otimização sobre um espaço infinitamente gerado é, aparentemente, impraticável. Entretanto, é recorrente em problemas de aprendizado de máquina a necessidade de solucionar uma certa classe de problemas que envolvem um mapeamento de vetores n dimensionais sobre um espaço de *Hilbert RKHS* denotado por $\mathcal{F}$. O Teorema 4 garante que o minimizador pertinente a essa classe de problemas é finitamente gerado, apesar de estar contido em $\mathcal{F}$.

Teorema 4 (Representante Semi-paramétrico[4])**.** *Sejam $\mathcal{F}$ um espaço RKHS com núcleo associado $K : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, S um conjunto formado pelos pares $(x_i, y_i) \in \mathbb{R}^{n+1}$ para $i = 1, \cdots, m$ e $\{\psi_j\}_{j=1,\cdots,M}$ um conjunto de funcionais lineares tais que a matriz*

$$[\psi_j(x_i)]_{\substack{1\leqslant i\leqslant m \\ 1\leqslant j\leqslant M}} = \begin{bmatrix} \psi_1(x_1) & \cdots & \psi_M(x_1) \\ \vdots & \cdots & \vdots \\ \psi_1(x_m) & \cdots & \psi_M(x_m) \end{bmatrix} \in \mathbb{R}^{m\times M}$$

possui posto M. Dessa forma, considere uma função monotonicamente crescente $\theta : [0, \infty) \mapsto \mathbb{R}$ e uma função convexa $\gamma : (\mathbb{R}^{n+2})^m \mapsto \mathbb{R} \cup \{\infty\}$. Então, quaisquer $f^* \in \mathcal{F}$ e $h^* \in \text{span}\{\psi_j\}$ tais que

$$(f^*, h^*) \in \arg \min_{\substack{f\in\mathcal{F} \\ h\in\text{span}\{\psi_j\}}} \Big\{\gamma\Big(\big(x_1, y_1, f(x_1) + h(x_1)\big), \cdots, \big(x_m, y_m, f(x_m) + h(x_m)\big)\Big) + \theta(||f||_{\mathcal{F}})\Big\} \tag{A.1}$$

são escritas como varreduras lineares finitas de $\mathcal{F}$ e $\text{span}\{\psi_j\}$, a saber

$$f^*(\cdot) = \sum_{i=0}^{m} \alpha_i K(\cdot, x_i)$$
$$h^*(\cdot) = \sum_{j=0}^{M} \beta_j \psi_j(\cdot),$$

em que $\alpha_i \in \mathbb{R}$ e $\beta_j \in \mathbb{R}$.

Demonstração. Seja a função $f \in \mathcal{F}$. Considere o espaço vetorial

$$V = \text{span}\{K(\cdot, x_1), \cdots, K(\cdot, x_m)\},$$

então a função f pode ser escrita como

$$f = \sum_{i=0}^{m} \alpha_i K(\cdot, x_i) + g,$$

[4] Teorema 4.3 de Schölkopf e Smola (2002).

em que $g \in V^{\perp}$.

Agora, como K é o núcleo associado a $\mathcal{F}$, observe que para qualquer x_j tal que $(x_j, y_j) \in S$, tem-se que

$$f(x_j) = \langle f, K(\cdot, x_j) \rangle_{\mathcal{F}} = \left\langle \sum_{i=0}^{m} \alpha_i K(\cdot, x_i) + g, K(\cdot, x_j) \right\rangle_{\mathcal{F}} = \left\langle \sum_{i=0}^{m} \alpha_i K(\cdot, x_i), K(\cdot, x_j) \right\rangle_{\mathcal{F}}.$$

Logo, a função γ independe de $V^{\perp}$, sendo determinada apenas por

$$\left\langle \sum_{i=0}^{m} \alpha_i K(\cdot, x_i), K(\cdot, x_j) \right\rangle_{\mathcal{F}} + h(x_j).$$

Por outro lado, utilizando o fato da função θ ser monotonicamente crescente, tem-se que

$$\begin{aligned} \theta(||f||_{\mathcal{F}}) &= \theta\left(\left\| \sum_{i=0}^{m} \alpha_i K(\cdot, x_i) + g \right\|_{\mathcal{F}} \right) = \theta\left(\left[\left\| \sum_{i=0}^{m} \alpha_i K(\cdot, x_i) \right\|_{\mathcal{F}}^2 + ||g||_{\mathcal{F}}^2 \right]^{\frac{1}{2}} \right) \\ \Rightarrow \quad \theta(||f||_{\mathcal{F}}) &\geqslant \theta\left(\left\| \sum_{i=0}^{m} \alpha_i K(\cdot, x_i) \right\|_{\mathcal{F}} \right). \end{aligned}$$

Portanto, note que tomar $f \in \mathcal{F}$ com $g = 0$ não afeta a função γ e decresce os valores computados por θ. Assim, qualquer f^* que minimiza (A.1) necessariamente é dada por

$$f^*(\cdot) = \sum_{i=0}^{m} \alpha_i K(\cdot, x_i).$$

Ademais, considere $h \in \text{span}\{\psi_j\}$. Assim, claramente h^* é escrita como

$$h^*(\cdot) = \sum_{j=0}^{M} \beta_j \psi_j(\cdot)$$

com $\beta_i \in \mathbb{R}$ unicamente determinada. De fato, note que computando h^* em S, tem-se que

$$[h^*(x_i)]_{x_i \in S} = \beta [\psi_j(x_i)]_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant M}}$$

em que $\beta \in \mathbb{R}^M$ representa os coeficientes β_i. Assim, utilizando o fato de $[\psi_j(x_i)]_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant M}}$ possuir posto M, então β é unicamente determinado e, por consequência, h^* também o é.

■