UNIVERSIDADE FEDERAL DO PARANÁ
Aline Cristiane Finkler

APRENDIZAGEM DE MÁQUINA APLICADA À PREVISÃO DOS MOVIMENTOS DO IBOVESPA

Curitiba
2017

UNIVERSIDADE FEDERAL DO PARANÁ
Aline Cristiane Finkler

APRENDIZAGEM DE MÁQUINA APLICADA À PREVISÃO DOS MOVIMENTOS DO IBOVESPA

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

Orientador: Prof. Dr. Geovani Nunes Grapiglia.

Curitiba
2017

F499a Finkler, Aline Cristiane
Aprendizagem de máquina aplicada à previsão dos movimentos do Ibovespa / Aline Cristiane Finkler. – Curitiba, 2017 .
97 f. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Matemática, 2017 .

Orientador: Geovani Nunes Grapiglia.

1. Indicadores Ibovespa. 2. Previsão dos Movimentos Ibovespa. 3. Estratégias de investimentos. I. Universidade Federal do Paraná. II. Grapiglia, Geovani Nunes. III. Título.

CDD: 511.66

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

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **ALINE CRISTIANE FINKLER** intitulada: **APRENDIZAGEM DE MÁQUINA APLICADA À PREVISÃO DOS MOVIMENTOS DO IBOVESPA**, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

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

Curitiba, 31 de Julho de 2017.

Geovani N Grapiglia

GEOVANI NUNES GRAPIGLIA

Presidente da Banca Examinadora (UFPR)

Lucas Garcia Pedroso

LUCAS GARCIA PEDROSO

Avaliador Interno (UFPR)

PAULO J. SILVA E SILVA

Avaliador Externo (Unicamp)

Dedico este Mestrado ao meu noivo e à minha família, pessoas que estão sempre ao meu lado, me apoiando em todos os momentos.

Agradecimentos

Agradeço ao meu orientador Prof. Dr. Geovani Nunes Grapiglia, por aceitar me orientar neste trabalho, exercendo esta função excelentemente, sem medir esforços para contribuir com minha formação acadêmica.

Aos incontáveis professores que fizeram parte da minha trajetória de estudante. Desde aqueles que me ensinaram a ler e escrever, assim como as mais básicas noções matemáticas, até os de nível universitário. Cada um deles teve sua importância para que eu chegasse até aqui.

Aos professores Paulo Silva (Unicamp) e Lucas Pedroso (UFPR), por concordarem em participar da banca de avaliação e acrescentarem enriquecedoras contribuições ao trabalho.

Aos meus pais Gilberto e Noeli, por tudo o que fizeram por mim ao longo de minha vida. Agradeço infinitamente pelo apoio incondicional e pela incansável dedicação a mim durante todos estes anos.

Aos meus irmãos, Ivan e Luana, por fazerem parte de minha vida, e me auxiliarem sempre que preciso.

Ao meu noivo Eduardo, por me apoiar e me incentivar em momentos difíceis. Mais do que isso, agradeço por aguentar ficar ao meu lado nos piores momentos. Além disso, obrigada pela disposição em me ajudar a crescer pessoal e profissionalmente.

À todos os amigos e familiares, por compreenderem minha ausência em certos momentos.

À Deus, por estar sempre presente.

Aos colegas, que foram prestativos em momentos de necessidade, e que contribuíram com um ambiente de estudos agradável.

Finalmente, à CAPES e Fundação Araucária, pelo apoio financeiro concedido.

"Os computadores são
incrivelmente rápidos,
precisos e burros;
os homens são
incrivelmente lentos,
imprecisos e brilhantes;
juntos, seus poderes
ultrapassam os limites
da imaginação."

Albert Einstein

Resumo

Neste trabalho investiga-se o uso de técnicas de Aprendizagem de Máquina para a previsão dos movimentos do Ibovespa, índice que representa o desempenho geral das ações negociadas na BM&FBovespa. Especificamente, são considerados modelos de Regressão Linear, Regressão Logística, C-SVM e Redes Neurais Artificiais. A partir de dados históricos mensais do Ibovespa, esses modelos são treinados para realizarem previsões binárias sobre o índice (de alta ou baixa), com horizontes de 1, 3, 6 e 12 meses. Nos testes realizados, com o modelo C-SVM chega-se a uma taxa de acerto de 72,7% para previsões de 6 meses. Essa taxa é melhorada para 78,8% usando-se um modelo que combina Regressão Linear, Regressão Logística e C-SVM. Tal modelo híbrido é então incorporado a uma estratégia de investimento com manutenção semestral para negociação do fundo de índices BOVA11, o qual busca replicar os movimentos do Ibovespa. Simulações sugerem que essa estratégia de investimento baseada em previsões é capaz de fornecer retornos significativamente maiores do que aqueles obtidos com uma estratégia simples conhecida como *buy and hold*. Esses resultados ilustram o grande potencial do uso de técnicas de aprendizagem de máquina como suporte para a tomada de decisões de compra e venda em bolsas de valores. Além disso, abordam-se aspectos teóricos referentes à alguns métodos de otimização. Em particular, um estudo unificado de complexidade para métodos de descida é apresentado.

Palavras-chave: *Otimização, Aprendizagem de Máquina, Ibovespa.*

Abstract

This work investigates the use of Machine Learning models for predicting the movements of Ibovespa, which is the index that represents the overall performance of the stocks negotiated in the BM&FBovespa. Specifically, the models considered are Linear Regression, Logistic Regression, C-SVM and Artificial Neural Networks. Using monthly data about Ibovespa, these models are trained to the task of predicting the index movements (up and down) for horizons of 1, 3, 6 and 12 months ahead. In the experiments performed, with a C-SVM it was possible to reach an accuracy of 72,7% for predictions 6 months ahead. This accuracy was improved up to 78,8% by using a suitable combination of Linear Regression, Logistic Regression and C-SVM. Then, this hybrid model was incorporated to a trading strategy for negotiation of index fund BOVA11, which tries to replicate the movements of Ibovespa. Numerical simulations suggest that this trading strategy based on forecasts is able to provide gains significantly higher than those obtained with a simple strategy known as buy and hold. These results illustrate the great potential of Machine Learning as support for trading decisions in the stock market. In addition, theoretical approaches to some optimization methods are discussed. In particular, a unified complexity study for descent methods is presented.

Keywords: *Optimization, Machine Learning, Ibovespa.*

Sumário

Lista de Figuras

Lista de Tabelas

Abreviaturas

AM	Aprendizagem de Máquina
BFGS	"Broyden-Fletcher-Goldfarb-Shanno"
DFP	"Davidon-Fletcher-Powell"
BM&FBovespa	Bolsa de Valores, Mercados e Futuros de São Paulo
Ibovespa	Índice Bovespa
ETF	"Exchange Traded Fund" (Fundo de Índice de Ações)
CDI	Certificado de Depósito Interbancário
CDB	Certificado de Depósito Bancário
LC	Letra de Câmbio
LCI	Letra de Crédito Imobiliário
LCA	Letra de Crédito do Agronegócio
RNA	Rede Neural Artificial
SVM	"Support Vector Machines" (Máquinas de Vetor Suporte)
C-SVM	Máquinas de Vetor Suporte com margens flexíveis
SEQ	Soma dos Erros Quadráticos
EC	Entropia Cruzada
TA	Taxa de Acerto

Notação

$\mathbb{N}$	Conjunto dos números naturais
$\mathbb{R}$	Conjunto dos números reais
$\mathbb{R}_{++}$	Números reais estritamente positivos
$\mathbb{R}^n$	Espaço euclidiano n-dimensional
$\mathbb{R}^{m\times n}$	Conjunto das matrizes reais com m linhas e n colunas
$\lvert\cdot\rvert$	Valor absoluto
$\lVert\cdot\rVert$	Norma euclidiana vetorial ou matricial
$\rho(X)$	Raio Espectral da matriz X
$B(x,\delta)$	Bola aberta de centro x e raio δ
$\lceil\cdot\rceil$	Número inteiro imediatamente superior ao número real no argumento
$\lfloor\cdot\rfloor$	Número inteiro imediatamente inferior ao número real no argumento
$\nabla f(x)$	Gradiente da função f no ponto x
$\nabla^2 f(x)$	Matriz Hessiana da função f no ponto x
$J_F(x)$	Jacobiana da função F no ponto x
$\lambda_{min}(A)$	Menor autovalor de A (em módulo)
$\lambda_{max}(A)$	Maior autovalor de A (em módulo)
$A \geq 0$	Matriz A simétrica definida positiva
$A \preceq B$	$B - A \geq 0$
$\text{cond}(A)$	Número de condição da matriz A
$\mathcal{C}^k$	Derivadas até ordem k contínuas
$\mathcal{O}(\varepsilon)$	Múltiplo de ε
I	Matriz Identidade
$proj_x(y)$	Projeção ortogonal de y em x
$d(X,Y)$	Distância entre X e Y
$x^{(i)}$	Vetores em $\mathbb{R}^n$ (entradas para os modelos de AM)
$y^{(i)}$	Valores em $\mathbb{R}$ (saídas para os modelos de AM)

Introdução

A BM&FBovespa (Bolsa de Valores, Mercados e Futuros) é a principal bolsa do Brasil, e uma das mais importantes da América Latina. Por meio dela, é possível comprar e vender ações, por meio eletrônico. A negociação no mercado de ações é uma forma de investimento que vem se tornando cada vez mais popular no Brasil. No processo de tomada de decisões de compra e venda, séries históricas dos preços das ações estão entre as informações mais importantes para os investidores. Evidentemente, a possibilidade de se prever os movimentos do preço de uma ação (alta ou baixa) é de grande interesse, uma vez que esse tipo de informação pode subsidiar decisões, tendo em vista a maximização de lucros ou a minimização de eventuais perdas decorrentes de oscilações do mercado. No entanto, dado o caráter dinâmico dessas séries temporais, a realização de previsões desse tipo com alto grau de acerto é uma tarefa extremamente difícil, gerando inclusive discussões teóricas sobre a sua viabilidade. Esse comportamento caótico ocorre pelo fato dos preços das ações serem afetados por diversos fatores sociais, políticos e macro-econômicos.

Recentemente, diversas técnicas de aprendizagem de máquina têm sido usadas com relativo sucesso na modelagem e previsão dos movimentos de preços em mercados de ações. Por exemplo, Dai e Zhang [9] utilizaram Regressão Logística e Máquinas de Vetor Suporte (SVM, do inglês Support Vector Machine) para obter previsões sobre uma única ação do mercado norte americano, a 3M. Embora os resultados para previsão de curto prazo (com horizonte de 1 a 7 dias) não tenham sido satisfatórios, para um horizonte de 44 dias eles conseguiram uma taxa de acerto de 79% utilizando SVM. Shen et al [26] utilizaram como variável explicativa uma variedade de dados mundiais (tais como índices de diversos mercados de ações, cotações de diferentes moedas, e ainda commodities como ouro e prata), e com um modelo SVM eles conseguiram prever os movimentos de alguns índices do mercado norte-americano (como o NASDAQ e S&P500) com uma taxa de acerto superior a 70% para o horizonte de 1 dia, e de até 85% para o horizonte de 30 dias. Huang et al [16] investigaram a eficiência da técnica SVM na previsão do movimento semanal do índice japonês NIKKEI 225, comparando-a com outras técnicas de classificação, tais como Redes Neurais Artificiais. Também neste caso, individualmente o modelo SVM teve performance superior, com taxa de acerto de 73% para previsão de 1 semana. No entanto, com uma combinação dos modelos eles obtiveram resultados ainda melhores, resultando em uma taxa de acerto de 75%. Majumder et al [21] utilizaram Redes Neurais Artificiais

para previsão dos movimentos do índice S&P CNX Nifty 50. Realizaram testes com diversas variações do modelo, obtendo uma taxa de acerto de 89.65%.

Motivada pela escassez de estudos desse gênero sobre o mercado de ações brasileiro, a presente dissertação tem como objetivo a aplicação de modelos de aprendizagem de máquina para a previsão do Ibovespa, que é o índice da BM&FBovespa que busca indicar o desempenho geral das ações registradas nessa bolsa de valores. Especificamente, são considerados os modelos de Regressão Linear, Regressão Logística, C-SVM e Redes Neurais Artificiais. Com o objetivo de melhorar o desempenho nas previsões, a técnica de combinação de modelos descrita em [16] também é investigada. Por fim, para ilustrar os potenciais ganhos decorrentes do uso desses modelos, várias simulações são realizadas comparando-se uma estratégia baseada em aprendizagem de máquina com uma estratégia simples do tipo *buy and hold*.

O restante do trabalho está organizado da seguinte maneira. O Capítulo 1 apresenta noções básicas de otimização e também uma descrição dos métodos de otimização usados para treinar os modelos abordados na dissertação. Em particular, os métodos de descida são apresentados de uma maneira unificada tendo como foco um estudo geral da complexidade de pior-caso desses métodos. O Capítulo 2 apresenta uma descrição detalhada dos modelos de aprendizagem de máquina considerados. O Capítulo 3 reporta os resultados de testes numéricos realizados na tentativa de se identificar o melhor modelo para a tarefa de previsão do Ibovespa. Por fim, no Capítulo 4 investiga-se, por meio de simulações, a aplicação do modelo mais eficiente descrito no Capítulo 3 como base para uma estratégia de investimento.

Capítulo 1

Noções de Otimização

Este capítulo contém noções básicas de otimização, bem como uma breve descrição dos métodos utilizados para resolver os problemas abordados na dissertação. As principais referências consideradas são Ribeiro e Karas [25], Luenberger e Ye [19] e Nesterov [23].

1.1 Definições e Resultados Básicos

Considere uma função $f : \mathbb{R}^n \to \mathbb{R}$ e um subconjunto $\Omega \subset \mathbb{R}^n$.

Definição 1.1. *Dado um ponto $x^* \in \Omega$,*

(a) diz-se que $x^* \in \Omega$ é minimizador global de f em Ω quando

$$f(x^*) \leq f(x), \quad \forall\, x \in \Omega;$$

(b) diz-se que $x^* \in \Omega$ é minimizador local de f em Ω quando existe $\delta > 0$ tal que

$$f(x^*) \leq f(x), \quad \forall\, x \in B(x^*, \delta) \cap \Omega.$$

O problema de minimização consiste em encontrar os minimizadores da função f no conjunto Ω, e pode ser escrito como

$$\begin{aligned} \min \quad & f(x) \\ s.a. \quad & x \in \Omega. \end{aligned} \tag{1.1}$$

A função $f : \mathbb{R}^n \to \mathbb{R}$ é denominada função objetivo, e $\Omega \subset \mathbb{R}^n$ é o conjunto viável.

Observação 1.2. *Todo problema de maximização*

$$\begin{aligned} \max \quad & f(x) \\ s.a. \quad & x \in \Omega \end{aligned} \tag{1.2}$$

pode ser transformado em um problema de minimização equivalente

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

Ambos os problemas (1.1) e (1.2) são referidos como problemas de otimização. Quando $\Omega = \mathbb{R}^n$, tem-se um problema de otimização irrestrito ou sem restrições. Quando $\Omega \subsetneq \mathbb{R}^n$, tem-se um problema de otimização com restrições. Neste caso, Ω costuma ser da forma

$$\Omega = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} r_i(x) = 0,\ i = 1, \ldots, p_e \\ r_i(x) \leq 0,\ i = p_e + 1, \ldots, p \end{array} \right\},$$

onde $p_e \leq p$ são inteiros não negativos, e $r_i : \mathbb{R}^n \to \mathbb{R}$, para cada $i = 1, \ldots, p$.

Os resultados abaixo fornecem condições suficientes para a existência de minimizadores globais.

Teorema 1.3 (Weierstrass)**.** *Sejam $f : \mathbb{R}^n \to \mathbb{R}$ contínua e $\Omega \subset \mathbb{R}^n$ compacto não vazio. Então existe minimizador global de f em Ω.*

Demonstração: Ver Teorema 2.2 em Ribeiro e Karas [25]. ∎

Corolário 1.4. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ contínua e suponha que existe $c \in \mathbb{R}$ tal que o conjunto $\mathcal{L} = \{x \in \mathbb{R}^n \mid f(x) \leq c\}$ é compacto não vazio. Então f tem um minimizador global.*

Demonstração: Ver Corolário 2.3 em Ribeiro e Karas [25]. ∎

No caso de problemas de otimização irrestritos, algumas condições devem ser satisfeitas para que um ponto $x^* \in \mathbb{R}^n$ seja minimizador.

Teorema 1.5 (Condição necessária de 1ª ordem)**.** *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função diferenciável. Se x^* é minimizador local de f, então*

$$\nabla f(x^*) = 0.$$

Demonstração: Ver Teorema 2.9 em Ribeiro e Karas [25]. ∎

Definição 1.6. *Um ponto x^* que satisfaz $\nabla f(x^*) = 0$ é chamado ponto crítico (ou estacionário) da função f.*

Segundo o Teorema 1.5, todo minimizador é um ponto crítico. No entanto, a recíproca nem sempre vale. Um caso importante onde todo ponto crítico é minimizador global ocorre quando f é uma função convexa.

Definição 1.7. *Uma função $f : \mathbb{R}^n \to \mathbb{R}$ é convexa quando*

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

para todos $x, y \in \mathbb{R}^n$ *e* $t \in (0,1)$.

Quando a desigualdade acima é estrita, dizemos que f *é estritamente convexa.*

Teorema 1.8. *Sejam* $f : \mathbb{R}^n \to \mathbb{R}$ *diferenciável. A função* f *é convexa se, e somente se,*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x), \tag{1.3}$$

para todos $x, y \in \mathbb{R}^n$.

Demonstração: Ver Teorema 3.13 em Ribeiro e Karas [25]. ∎

Corolário 1.9. *Se* $f : \mathbb{R}^n \to \mathbb{R}$ *é uma função convexa, qualquer ponto crítico é minimizador global de* f.

Demonstração: Como f é convexa, segue de (1.3) que

$$f(x) \geq f(x^*) + \nabla f(x^*)(x - x^*) \geq f(x^*), \quad \forall x \in \mathbb{R}^n,$$

pois $\nabla f(x^*) = 0$. ∎

Isto significa que para minimizar uma função convexa f, basta encontrar um ponto crítico, ou seja, um ponto x^* tal que $\nabla f(x^*) = 0$. Por conta desse resultado, em otimização é extremamente importante identificar quando uma função é convexa. Nesse contexto, convém revisar o conceito de matriz (semi)definida positiva.

Definição 1.10. *Uma matriz simétrica* $A \in \mathbb{R}^{n \times n}$ *é dita definida positiva quando*

$$x^T A x > 0$$

para todo $x \in \mathbb{R}^n \setminus \{0\}$. *Neste caso, escreve-se* $A > 0$.

Se

$$x^T A x \geq 0$$

para todo $x \in \mathbb{R}^n$, *diz-se que* A *é semidefinida positiva, e denota-se por* $A \geq 0$.

Teorema 1.11. *Seja* $f : \mathbb{R}^n \to \mathbb{R}$ *uma função de classe* $\mathcal{C}^2$. *Se* $\nabla^2 f(x) \geq 0$ *para todo* $x \in \mathbb{R}^n$, *então* f *é convexa. A recíproca também é válida.*

Demonstração: Ver Teorema 3.16 em Ribeiro e Karas [25]. ∎

Considerando informações de segunda ordem da função, tem-se a seguinte condição necessária de otimalidade:

Teorema 1.12 (Condição necessária de 2ª ordem)**.** *Seja* $f : \mathbb{R}^n \to \mathbb{R}$ *uma função duas vezes diferenciável. Se* x^* *é minimizador local de* f, *então*

$$\nabla^2 f(x^*) \geq 0.$$

Demonstração: Ver Teorema 2.12 em Ribeiro e Karas [25]. ∎

Informações de segunda ordem também permitem identificar quando um ponto crítico é um minimizador local de f.

Teorema 1.13 (Condição suficiente de 2ª ordem)**.** *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função duas vezes diferenciável. Se*

$$\nabla f(x^*) = 0 \quad e \quad \nabla^2 f(x^*) > 0,$$

então x^* é minimizador local estrito de f.

Demonstração: Ver Teorema 2.14 em Ribeiro e Karas [25]. ∎

Os resultados acima referentes a minimizadores podem ser facilmente adaptados para maximizadores. Entretanto, existem pontos críticos que não são nem maximizadores nem minimizadores.

Definição 1.14. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função diferenciável e $\bar{x}$ um ponto crítico de f. Diz-se que $\bar{x}$ é ponto de sela de f quando, para todo $\delta > 0$, existem $x, y \in B(\bar{x}, \delta)$ tais que*

$$f(x) < f(\bar{x}) < f(y).$$

A identificação de pontos de sela pode ser feita a partir da noção de matriz indefinida.

Definição 1.15. *Uma matriz simétrica $A \in \mathbb{R}^{n\times n}$ é dita indefinida quando existem $x, y \in \mathbb{R}^n$ tais que*

$$x^T Ax < 0 < y^T Ay.$$

Teorema 1.16. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função duas vezes diferenciável no ponto estacionário $\bar{x}$. Se $\nabla^2 f(\bar{x})$ é indefinida, então $\bar{x}$ é ponto de sela de f.*

Demonstração: Ver Teorema 2.16 em Ribeiro e Karas [25]. ∎

1.2 Dedução de Métodos de Descida

Muitos problemas práticos podem ser reduzidos à busca por um vetor $x \in \mathbb{R}^n$ tal que

$$F(x) = 0, \tag{1.4}$$

onde $F : \mathbb{R}^n \to \mathbb{R}^n$ é uma função diferenciável não linear.

Na maioria das vezes, resolver tal problema de maneira direta é muito complicado. Por isso, recorre-se a métodos iterativos, os quais geram uma sequência (x_k) de aproximações. Dada uma aproximação x_k para a solução de (1.4), o ideal seria encontrar um passo $d_k \in \mathbb{R}^n$ tal que

$$F(x_k + d_k) = 0.$$

Ora, sendo F diferenciável, tem-se que

$$F(x_k + d) = F(x_k) + J_F(x_k)d + r(d),$$

onde $J_F(x_k)$ é a Jacobiana de F em x_k, e $\lim_{\|d\|\to 0} \frac{r(d)}{\|d\|} = 0$. Em particular, $\lim_{\|d\|\to 0} r(d) = 0$. Assim, para $\|d\|$ suficientemente pequena, obtém-se

$$F(x_k + d) \cong F(x_k) + J_F(x_k)d. \tag{1.5}$$

A relação (1.5) sugere a busca por um passo d_k tal que

$$F(x_k) + J_F(x_k)d_k = 0. \tag{1.6}$$

Se $J_F(x_k)$ é não singular, a solução de (1.6) é

$$d_k = -J_F(x_k)^{-1}F(x_k).$$

Deste modo, obtém-se a seguinte regra para atualização de x_k:

$$x_{k+1} = x_k + d_k = x_k - J_F(x_k)^{-1}F(x_k). \tag{1.7}$$

O processo iterativo (1.7) é conhecido como *Método de Newton Puro*.

A Figura 1.1 ilustra uma interpretação geométrica do método para $n = 1$, onde procura-se aproximar as raízes de $F(x) = 2x^2 + x + 1$. Neste caso, tem-se $J_F(x) = F'(x)$. Esta figura sugere que a sequência (x_k) gerada pelo Método de Newton converge para uma solução x^* do problema (1.4).

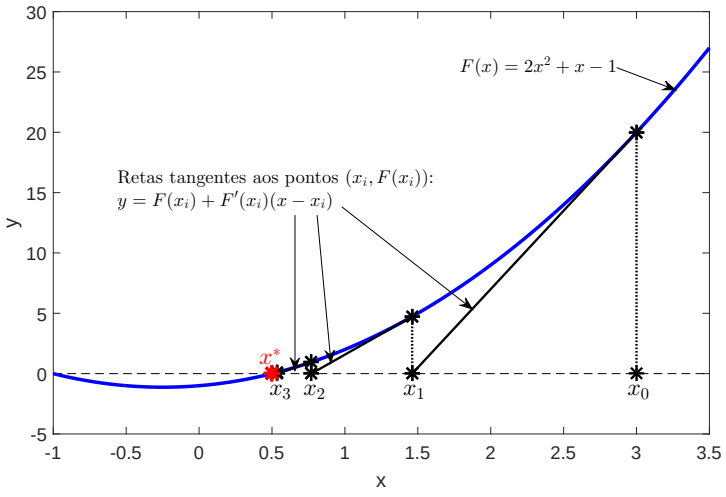


Figura 1.1: Interpretação geométrica do Método de Newton.
O ponto x_{k+1} é resultado da interseção da reta tangente ao gráfico de F no ponto $(x_k, F(x_k))$ com o eixo x.

Infelizmente, se o ponto inicial x_0 não estiver suficientemente próximo de x^*, a sequência (x_k) gerada pelo Método de Newton Puro pode divergir. Por exemplo, considere a função $F : \mathbb{R} \to \mathbb{R}$ dada por

$$F(x) = \frac{x}{\sqrt{1+x^2}}.$$

Para essa função, o Método de Newton diverge quando $|x_0| \geq 1$. De fato, a solução de $F(x) = 0$ é $x^* = 0$, e

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)} = x_k - \frac{x_k(1+x_k^2)^{-\frac{1}{2}}}{(1+x_k^2)^{-\frac{3}{2}}} = -x_k^3.$$

Se $|x_0| < 1$, a sequência (x_k) converge rapidamente para x^*, uma vez que $|x_k^3| \to 0$. Se $x_0 = 1$, tem-se $x_1 = -1$ e $x_2 = x_0 = 1$, de modo que o método entra num processo infinito, e nunca encontra a raiz $x^* = 0$. Se $|x_0| > 1$, então $|x_k| \to \infty$. Este exemplo está ilustrado na Figura 1.2, onde $x_0 = 1$ é tomado como ponto inicial.

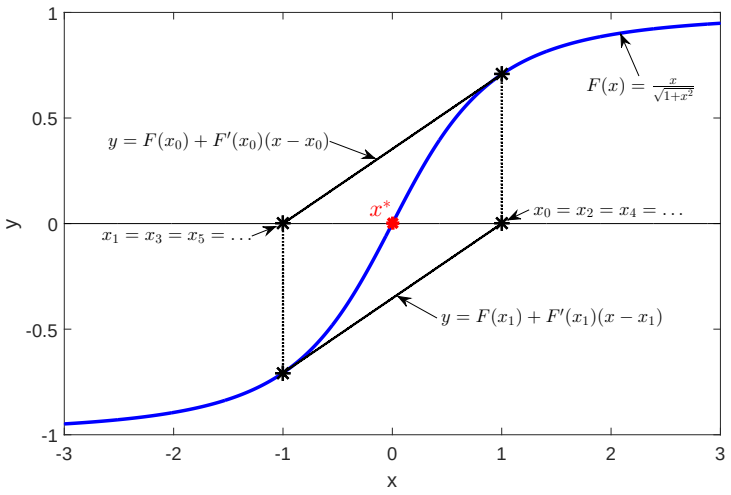


Figura 1.2: Exemplo de divergência do Método de Newton Puro.

Uma forma de se contornar este problema é o controle do tamanho do passo a partir de uma sequência $(t_k) \subset \mathbb{R}_{++}$. Especificamente, o Método de Newton Puro é modificado da seguinte maneira:

$$x_{k+1} = x_k + t_k d_k = x_k - t_k J_F(x_k)^{-1} F(x_k). \tag{1.8}$$

O processo iterativo (1.8) é conhecido como *Método de Newton com Busca*, pois ele depende da busca dos parâmetros t_k de modo a garantir a convergência do método para qualquer ponto inicial x_0.

Por outro lado, mesmo no Método de Newton com Busca, $J_F(x_k)$ pode ser singular, tornando a sequência (x_k) mal-definida. Este problema pode ser evitado substituindo-se $J_F(x_k)$ por uma matriz não singular $H_k \in \mathbb{R}^{n \times n}$. Para preservar as propriedades do

método, é interessante que tal matriz H_k seja uma aproximação de $J_F(x_k)$ tão boa quanto possível. Denotando $B_k = H_k^{-1}$, a partir de (1.8) obtém-se o seguinte processo iterativo:

$$x_{k+1} = x_k + t_k d_k, \text{ com } d_k = -B_k F(x_k). \tag{1.9}$$

Em otimização suave irrestrita, dada uma função $f : \mathbb{R}^n \to \mathbb{R}$, o objetivo é encontrar um minimizador x^* de f. Neste caso, sabe-se que se x^* é minimizador de f, então $\nabla f(x^*) = 0$ (Teorema 1.5). Assim, na prática busca-se por uma solução da equação não linear

$$\nabla f(x) = 0.$$

Para $F(x) = \nabla f(x)$, o esquema (1.9) se torna

$$x_{k+1} = x_k + t_k d_k, \text{ com } d_k = -B_k \nabla f(x_k). \tag{1.10}$$

Aqui, B_k atua como uma aproximação para $\nabla^2 f(x_k)^{-1}$, e geralmente considera-se B_k como sendo simétrica e definida positiva. Quando f é de classe $\mathcal{C}^2$, segue do Teorema de Schwarz[1] que $\nabla^2 f(x_k) \in \mathbb{R}^{n\times n}$ é simétrica. Isto justifica a simetria de B_k. O fato de B_k ser definida positiva é motivado pela seguinte propriedade:

Teorema 1.17. *Seja B uma matriz simétrica e definida positiva. Se $d = -B\nabla f(x)$, então*

$$f(x + td) < f(x) \tag{1.11}$$

para todo t suficientemente pequeno.

Observação 1.18. *Se $d \in \mathbb{R}^n$ satisfaz* (1.11)*, diz-se que d é uma direção de descida. Para que uma direção d seja de descida a partir de um ponto x, é suficiente que se tenha $d^T \nabla f(x) < 0$ (Ver Teorema 4.2 em Ribeiro e Karas [25]).*

Em razão do Teorema 1.17, o método (1.10) será chamado de *Método Geral de Descida*. Este método pode ser descrito da seguinte maneira:

[1] Ver Teorema 4 do Capítulo 3.3 de Lima [17].

Algoritmo 1.1. *Método de Descida*

Passo 0: Dados $x_0 \in \mathbb{R}^n$ *e* $B_0 \in \mathbb{R}^{n\times n}$ *simétrica e definida positiva, defina* $k = 0$.

Passo 1: Se $\nabla f(x_k) = 0$*, pare;*

Passo 2: Calcule $d_k = -B_k \nabla f(x_k)$*;*

Passo 3: Calcule $t_k > 0$ *tal que* $f(x_k + t_k d_k) < f(x_k)$*;*

Passo 4: Defina $x_{k+1} = x_k + t_k d_k$*;*

Passo 5: Escolha B_{k+1} *simétrica e definida positiva;*

Passo 6: Defina $k = k + 1$ *e volte ao Passo 1.*

O cálculo de t_k no Passo 3 pode ser feito de diversas maneiras. Entre elas, destacam-se as seguintes:

- Busca Exata: consiste em tomar t_k como a solução do problema

$$\min_{t\in\mathbb{R}_+} \quad \phi(t) = f(x + td). \tag{1.12}$$

  Mesmo sendo unidimensional, o problema (1.12) pode ser bastante complicado. Nestes casos, métodos de busca inexata podem ser mais viáveis.

- Busca Inexata de Goldstein-Armijo: consiste em encontrar um $t_k > 0$ de modo que haja uma redução no valor da função na direção d_k, sem necessidade de resolver o problema (1.12). Para isto, utiliza-se a regra

$$0 < -\mu_1 t_k \nabla f(x_k)^T d_k \leq f(x_k) - f(x_k + t_k d_k) \leq -\mu_2 t_k \nabla f(x_k)^T d_k,$$

  onde $0 < \mu_1 < \mu_2 < 1$.

- Passo constante: consiste em fazer $t_k = t$ para todo k. Esta técnica costuma ser pouco eficiente, uma vez que o mesmo tamanho de passo deve garantir o decréscimo da função na direção escolhida para qualquer ponto de partida, o que pode significar que o passo seja muito pequeno, tornando o algoritmo lento.

1.3 Análise Teórica de Métodos de Descida

Dado um problema de otimização, uma solução x^* do mesmo geralmente satisfaz uma condição de criticalidade da forma $\mu(x^*) = 0$, com $\mu(x) \geq 0$. Quando x satisfaz $\mu(x) = 0$, diz-se que x é um ponto crítico do problema. No caso de um problema de minimização suave sem restrições, a medida de criticalidade usual é $\mu(x) = \|\nabla f(x)\|$, onde f é a

função objetivo. Outra medida válida, mas nem sempre viável de ser calculada, é $\mu(x) = f(x) - f^*$, onde f^* é o valor mínimo de f.

Por conta da precisão finita dos computadores, implementações práticas de métodos de otimização não utilizam a condição $\mu(x_k) = 0$ como critério de parada. Em vez disso, elas fazem uso da condição mais fraca

$$\mu(x_k) \leq \varepsilon, \tag{1.13}$$

onde $\varepsilon > 0$ é uma tolerância fixada *a priori* pelo usuário. Para um método iterativo de otimização, é extremamente desejável que se tenha a garantia teórica de que o critério de parada (1.13) será satisfeito, independentemente da escolha do ponto inicial x_0. Essa propriedade é conhecida como **Convergência Global**, e pode ser formalizada da seguinte maneira:

Definição 1.19 (Convergência Global)**.** *Seja (x_k) a sequência gerada por um método iterativo de otimização M a partir de um ponto inicial arbitrário x_0. Diz-se que o método M é globalmente convergente quando, dado $\varepsilon > 0$, existe $\bar{k} = \bar{k}(\varepsilon, x_0) \in \mathbb{N}$ tal que $\mu(x_{\bar{k}}) \leq \varepsilon$.*

O exemplo de divergência do Método de Newton Puro descrito na Seção 1.2 mostra que nem todo método de otimização é globalmente convergente.

Apesar de ser uma propriedade importante, a convergência global não diz muito sobre a eficiência de um método de otimização. Ela apenas garante que a execução do método vai parar em algum momento. Obviamente, é desejável que o método seja rápido, ou seja, que ele pare executando o menor número possível de iterações. Assim, para avaliar a eficiência é interessante estimar o quão grande é o primeiro $\bar{k}$ para o qual $\mu(x_{\bar{k}}) \leq \varepsilon$, isto é, o número máximo de iterações que o método precisa executar no pior caso até que o critério (1.13) seja satisfeito. Fixado x_0, quanto menor for $\varepsilon > 0$, maior será $\bar{k}$. Esse tipo de limitante superior sobre $\bar{k}$ caracteriza a **Complexidade de Pior-Caso** do método.

Em geral, limitantes de complexidade são da forma $\bar{k} \leq \mathcal{O}(\varepsilon^{-p})$, com $p \in \{1, 1.5, 2\}$. Mesmo para $p = 1$, se consideramos $\varepsilon = 10^{-6}$ obtemos um limitante superior da ordem de um milhão de iterações. Felizmente, na prática quase sempre é possível satisfazer o critério de parada (1.13) com um número de iterações muito menor que o número previsto pela análise de complexidade.

Uma outra medida de eficiência, menos pessimista, é a **Taxa de Convergência**. A análise da taxa de convergência permite avaliar a velocidade de convergência das sequências geradas por um método quando as suas iteradas estão suficientemente próximas de uma solução do problema. A definição a seguir apresenta caracterizações para a taxa de convergência de uma sequência.

Definição 1.20. *Seja (x_k) uma sequência que converge para x^*, com $x_k \neq x^*$, para todo k. Diz-se que a convergência de (x_k) é:*

- *Linear, com taxa de convergência r, quando*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = r < 1;$$

- *Superlinear, quando*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0;$$

- *Sublinear, quando*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 1;$$

- *de ordem $p > 1$, quando*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} < \infty;$$

  Quando $p = 2$, diz-se que a convergência é quadrática.

Observação 1.21. *Quanto maior a ordem de convergência das sequências geradas por um método, mais rápido ele tende a ser. Assim, é preferível um método com convergência quadrática do que linear, por exemplo.*

A seguir, o Algoritmo 1.1 é analisado tendo como foco esses três aspectos fundamentais: Convergência Global, Complexidade de Pior-Caso e Taxas de Convergência.

1.3.1 Convergência Global e Complexidade de Pior-Caso

São discutidas agora a ordem de complexidade e a convergência global do Método de Descida descrito no Algoritmo 1.1.

Definição 1.22. *Uma função $f : X \subset \mathbb{R}^n \to \mathbb{R}^m$ é Lipschitziana se existe uma constante $L > 0$ tal que*

$$\|f(x) - f(y)\| \leq L \|x - y\|,$$

para todos $x, y \in X$. Neste caso, diz-se que f é L-Lipschitz.

Considere as seguintes hipóteses:

(H1) $f : \mathbb{R}^n \to \mathbb{R}$ é diferenciável, e $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ é L-Lipschitz.

(H2) Existem constantes positivas $c_0 \leq c_1$ tais que, para todo k, B_k é simétrica, e

$$c_0 I \preceq B_k \preceq c_1 I.$$

Observação 1.23. *Note que (H2) implica que $\lambda_{min}(B_k) \geq c_0$ e $\|B_k\| = \lambda_{max}(B_k) \leq c_1$, para todo k.*

Lema 1.24. *Suponha que (H1) e (H2) sejam satisfeitas. Dado $x_0 \in \mathbb{R}^n$, seja $(x_k) \subset \mathbb{R}^n$ a sequência gerada pelo Algoritmo 1.1 a partir de x_0. Então, para todo k*

$$f(x_k) - f(x_{k+1}) \geq t_k \left(c_0 - \frac{L c_1^2}{2} t_k \right) \|\nabla f(x_k)\|^2 . \tag{1.14}$$

Demonstração: Como ∇f é L-Lipschitz, sabe-se que[2]

$$\left| f(y) - f(x) - \nabla f(x)^T (y - x) \right| \leq \frac{L}{2} \|y - x\|^2 , \quad \forall\, x, y \in \mathbb{R}^n. \tag{1.15}$$

Fazendo $y = x_{k+1}$ e $x = x_k$, tem-se

$$f(x_{k+1}) - f(x_k) - \nabla f(x_k)^T (x_{k+1} - x_k) \leq \frac{L}{2} \|x_{k+1} - x_k\|^2 . \tag{1.16}$$

Observe que

$$\begin{aligned} \nabla f(x_k)^T (x_{k+1} - x_k) &= -t_k \nabla f(x_k)^T B_k \nabla f(x_k) \\ &\leq -t_k \lambda_{min}(B_k) \|\nabla f(x_k)\|^2 \end{aligned} \tag{1.17}$$

Por outro lado,

$$\begin{aligned} \frac{L}{2} \|x_{k+1} - x_k\|^2 &= \frac{L}{2} \|t_k B_k \nabla f(x_k)\|^2 \\ &\leq \frac{L}{2} t_k^2 \|B_k\|^2 \|\nabla f(x_k)\|^2 \\ &\leq \frac{L}{2} t_k^2 \left(\lambda_{max}(B_k)\right)^2 \|\nabla f(x_k)\|^2 \end{aligned} \tag{1.18}$$

Combinando (1.16), (1.17) e (1.18), tem-se que

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\leq \frac{L}{2} t_k^2 \left(\lambda_{max}(B_k)\right)^2 \|\nabla f(x_k)\|^2 - t_k \lambda_{min}(B_k) \|\nabla f(x_k)\|^2 \\ &= -t_k \left(\lambda_{min}(B_k) - \frac{L}{2} \left(\lambda_{max}(B_k)\right)^2 t_k \right) \|\nabla f(x_k)\|^2 . \end{aligned}$$

Logo,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq t_k \left(\lambda_{min}(B_k) - \frac{L}{2} \left(\lambda_{max}(B_k)\right)^2 t_k \right) \|\nabla f(x_k)\|^2 \\ &\geq t_k \left(c_0 - \frac{L}{2} c_1^2 t_k \right) \|\nabla f(x_k)\|^2 . \end{aligned}$$

■

Observação 1.25. *A melhor estimativa que se pode obter para a desigualdade* (1.14) *é*

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L} \left(\frac{\lambda_{min}(B_k)}{\lambda_{max}(B_k)} \right)^2 \|\nabla f(x_k)\|^2 \geq \frac{1}{2L} \frac{c_0^2}{c_1^2} \|\nabla f(x_k)\|^2 .$$

[2]Ver Lema 1.2.3 em Nesterov [23].

De fato, defina

$$h(t) = t\left(c_0 - \frac{L}{2}c_1^2 t\right).$$

Ao maximizar a função h (o que é equivalente a minimizar $-h$), obtém-se o maior valor que a função $h : \mathbb{R} \to \mathbb{R}$ pode assumir. Como $-h(t)$ é uma função convexa, seu minimizador é solução da equação

$$-h'(t) = Lc_1^2 t - c_0 = 0.$$

Portanto, tem-se que

$$t^* = \frac{c_0}{Lc_1^2}$$

é maximizador de h, e consequentemente

$$h(t^*) = \frac{c_0^2}{2Lc_1^2}$$

é o máximo de h.

Observação 1.26. *Note que a estimativa de decréscimo da função f na k-ésima iteração está relacionada com o número de condição da matriz B_k. Como*

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\left(\frac{\lambda_{min}(B_k)}{\lambda_{max}(B_k)}\right)^2 \|\nabla f(x_k)\|^2 = \frac{1}{2L}\left(cond(B_k)\right)^{-2} \|\nabla f(x_k)\|,$$

tem-se que quanto menor o número de condição da matriz B_k, maior tende a ser o decréscimo de f.

Lema 1.27. *Suponha que (H1) e (H2) sejam satisfeitas. Dado $x_0 \in \mathbb{R}^n$, seja $(x_k) \subset \mathbb{R}^n$ a sequência gerada pelo Algoritmo 1.1 a partir de x_0. Considere os seguintes casos para o cálculo de t_k:*

(a) $t_k = \alpha\dfrac{2c_0}{Lc_1^2}$, com $\alpha \in (0,1)$ (passo constante);

(b) t_k é obtido pela Busca de Armijo, satisfazendo

$$f(x_k) \geq f(x_k + t_k d_k) - \mu t_k \nabla f(x_k)^T d_k, \tag{1.19}$$

com $\mu \in (0,1)$;

(c) t_k é obtido pela Busca de Goldstein-Armijo, satisfazendo

$$0 < -\mu_1 t_k \nabla f(x_k)^T d_k \leq f(x_k) - f(x_k + t_k d_k) \leq -\mu_2 t_k \nabla f(x_k)^T d_k,$$

com $0 < \mu_1 < \mu_2 < \frac{c_0}{c_1}$.

Então, nestes casos existe $w > 0$ tal que, para todo k,

$$f(x_k) - f(x_{k+1}) \geq \frac{w}{L} \|\nabla f(x_k)\|^2,$$

onde

$$w = \begin{cases} \dfrac{2c_0^2}{c_1^2}\alpha(1-\alpha), & \textit{no caso (a)} \\ 2\mu\dfrac{c_0^2}{c_1^2}, & \textit{no caso (b)} \\ 2\mu_1 c_0 \dfrac{c_0 - \mu_2 c_1}{c_1^2}, & \textit{no caso (c)} \end{cases} \tag{1.20}$$

Demonstração:

(a) Neste caso, tem-se

$$t = \alpha\frac{2c_0}{Lc_1^2}, \quad \forall\, k \tag{1.21}$$

com $\alpha \in (0,1)$. Então, substituindo (1.21) em (1.14) (Lema 1.24) segue-se que

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \alpha\frac{2c_0}{Lc_1^2}\left(c_0 - \frac{L}{2}\alpha\frac{2c_0}{Lc_1^2}c_1^2\right)\|\nabla f(x_k)\|^2 \\ &= \alpha\frac{2c_0}{Lc_1^2}\left(c_0 - \alpha c_0\right)\|\nabla f(x_k)\|^2 \\ &= \frac{2c_0^2}{Lc_1^2}\alpha\left(1-\alpha\right)\|\nabla f(x_k)\|^2 \\ &= \frac{w}{L}\|\nabla f(x_k)\|^2, \end{aligned}$$

para

$$w = \frac{2c_0^2}{c_1^2}\alpha\left(1-\alpha\right).$$

(b) Pelo Lema 1.24 tem-se que

$$f(x_k) - f(x_{k+1}) \geq t_k\left(c_0 - \frac{Lc_1^2}{2}t_k\right)\|\nabla f(x_k)\|^2 \geq 0 \quad \Rightarrow \quad t_k \leq \frac{2c_0}{Lc_1^2}.$$

Tendo isto, como t_k satisfaz a condição de Armijo dada em (1.19), obtém-se

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq -\mu t_k \nabla f(x_k)^T d_k \\ &\geq -\mu\frac{2c_0}{Lc_1^2}\nabla f(x_k)^T d_k \\ &= \mu\frac{2c_0}{Lc_1^2}\nabla f(x_k)^T B_k \nabla f(x_k) \\ &\geq \mu\frac{2c_0^2}{Lc_1^2}\|\nabla f(x_k)\|^2. \end{aligned}$$

Assim, se t_k satisfaz a condição de Armijo, segue que

$$f(x_k) - f(x_{k+1}) \geq \frac{w}{L} \|\nabla f(x_k)\|^2,$$

onde

$$w = 2\mu \frac{c_0^2}{c_1^2}.$$

(c) Neste caso, t_k satisfaz as seguintes desigualdades:

$$f(x_k) - f(x_{k+1}) \geq \mu_1 t_k \nabla f(x_k)^T B_k \nabla f(x_k) \geq \mu_1 t_k c_0 \|\nabla f(x_k)\|^2 \tag{1.22}$$

e

$$f(x_k) - f(x_{k+1}) \leq \mu_2 t_k \nabla f(x_k)^T B_k \nabla f(x_k) \leq \mu_2 t_k c_1 \|\nabla f(x_k)\|^2, \tag{1.23}$$

com $\mu_2 < \dfrac{c_0}{c_1}$. Além disso, pelo Lema 1.24

$$f(x_k) - f(x_{k+1}) \geq t_k \left(c_0 - \frac{L}{2} c_1^2 t_k \right) \|\nabla f(x_k)\|^2. \tag{1.24}$$

Por (1.23) e (1.24), tem-se

$$\begin{aligned} \mu_2 t_k c_1 \|\nabla f(x_k)\|^2 \geq t_k \left(c_0 - \frac{L}{2} c_1^2 t_k \right) \|\nabla f(x_k)\|^2 &\Rightarrow \mu_2 c_1 \geq c_0 - \frac{L}{2} c_1^2 t_k \\ &\Rightarrow \frac{L}{2} c_1^2 t_k \geq c_0 - \mu_2 c_1 \\ &\Rightarrow t_k \geq \frac{2(c_0 - \mu_2 c_1)}{L c_1^2}. \end{aligned} \tag{1.25}$$

Utilizando (1.25) em (1.22), obtém-se que

$$f(x_k) - f(x_{k+1}) \geq \mu_1 \frac{2(c_0 - \mu_2 c_1)}{L c_1^2} c_0 \|\nabla f(x_k)\|^2,$$

ou ainda

$$f(x_k) - f(x_{k+1}) \geq \frac{2}{L} \mu_1 c_0 \left(\frac{c_0 - \mu_2 c_1}{c_1^2} \right) \|\nabla f(x_k)\|^2.$$

Assim, para t_k obtido através do método de Goldstein-Armijo tem-se

$$f(x_k) - f(x_{k+1}) \geq \frac{w}{L} \|\nabla f(x_k)\|^2,$$

com

$$w = 2\mu_1 c_0 \frac{c_0 - \mu_2 c_1}{c_1^2}.$$

■

Teorema 1.28. *Dado $x_0 \in \mathbb{R}^n$, seja (x_k) a sequência gerada pelo Algoritmo 1.1 a partir de x_0, onde para todo k, t_k é calculado pela busca (a) ou pela busca (b) descritas no Lema 1.27. Suponha que (H1) e (H2) são satisfeitas, e que o conjunto de nível $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ é compacto. Então, denotando*

$$g_k^* = \min_{i=1,\ldots,k-1} \|\nabla f(x_i)\|,$$

tem-se que

$$g_k^* \leq \left[\frac{L\,(f(x_0) - f^*)}{wk}\right]^{\frac{1}{2}}, \quad \forall\, k > 0,$$

onde f^* é o valor mínimo de f e a constante w é especificada em (1.20).
Consequentemente, dado $\varepsilon > 0$, o Método de Descida executa no máximo $\mathcal{O}(\varepsilon^{-2})$ iterações para gerar x_k tal que $\|\nabla f(x_k)\| \leq \varepsilon$.

Demonstração: Como $\mathcal{L}(x_0)$ é compacto, segue do Corolário 1.4 que f possui um valor mínimo f^*. Pelo Lema 1.27, sabe-se que existe $w > 0$ tal que

$$f(x_i) - f(x_{i+1}) \geq \frac{w}{L} \|\nabla f(x_i)\|^2, \quad \forall\, i \in \mathbb{N}.$$

Assim, dado $k > 0$ tem-se

$$f(x_0) - f(x_k) = \sum_{i=0}^{k-1} f(x_i) - f(x_{i+1}) \geq \frac{w}{L} \sum_{i=0}^{k-1} \|\nabla f(x_i)\|^2.$$

Logo,

$$f(x_0) - f^* \geq f(x_0) - f(x_k) \geq \frac{w}{L} k \left(\min_{i=1,\ldots,k-1} \|\nabla f(x_i)\|\right)^2 = \frac{w}{L} k\,(g_k^*)^2 \tag{1.26}$$

$$\Rightarrow \quad g_k^* \leq \left[\frac{L\,(f(x_0) - f^*)}{wk}\right]^{\frac{1}{2}}, \quad \forall\, k > 0.$$

Agora, seja $\bar{k}$ o menor índice para o qual $\|\nabla f(x_{\bar{k}})\| \leq \varepsilon$. Então, $\|\nabla f(x_i)\| > \varepsilon$, para $i = 0, \ldots, \bar{k} - 1$. Consequentemente, tem-se $g_{\bar{k}}^* > \varepsilon$ e, por (1.26),

$$f(x_0) - f^* \geq \frac{w}{L} \bar{k} \left(g_{\bar{k}}^*\right)^2 > \frac{w}{L} \bar{k} \varepsilon^2.$$

Portanto,

$$\bar{k} < \frac{L(f(x_0) - f^*)}{w\varepsilon^2}. \tag{1.27}$$

Observe que, de acordo com (1.27), são necessárias no máximo $\mathcal{O}(\varepsilon^{-2})$ iterações para se obter x_k tal que $\|\nabla f(x_k)\| \leq \varepsilon$. ∎

Corolário 1.29. *Considere as mesmas hipóteses do Teorema 1.28. Então, (x_k) possui*

uma subsequência que converge para um ponto crítico de f, ou seja, existe pelo menos um ponto de acumulação que é ponto crítico de f.

Demonstração: Segue do Teorema 1.28 que, dado $i \in \mathbb{N}$, existe $k_i \in \mathbb{N}$ tal que

$$0 \leq \|\nabla f(x_{k_i})\| \leq \frac{1}{i}.$$

Logo,

$$\lim_{i \to \infty} \|\nabla f(x_{k_i})\| = 0. \tag{1.28}$$

Como $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ é compacto, segue-se que $(x_{k_i}) \subset \mathcal{L}(x_0)$ é uma sequência limitada, e portanto tem uma subsequência convergente $(x_{k_{i_j}})$. Suponha que $x_{k_{i_j}} \to x^*$. Como ∇f é contínuo, tem-se que

$$\nabla f(x_{k_{i_j}}) \to \nabla f(x^*). \tag{1.29}$$

Combinando (1.28) com (1.29), pode-se concluir que

$$\nabla f(x^*) = 0.$$

Isto significa que pelo menos um ponto de acumulação de (x_k) é ponto crítico de f. ∎

Quando assume-se que a função objetivo possui certas propriedades adicionais, comumente os resultados tendem a ficar melhores ou mais simples. Um exemplo disso é o caso das funções convexas.

Teorema 1.30. *Considere as mesmas hipóteses do Teorema 1.28. Suponha ainda que a função objetivo f é convexa. Então, tem-se que*

$$f(x_k) - f^* \leq \frac{LD^2}{kw}, \quad \forall\, k > 0,$$

onde $D = sup\{\|x - y\| : x, y \in \mathcal{L}(x_0)\}$ é o diâmetro do conjunto $\mathcal{L}(x_0)$. Consequentemente, dado $\varepsilon \in (0, 1)$, o Método de Descida executa no máximo $\mathcal{O}(\varepsilon^{-1})$ iterações para gerar x_k tal que $f(x_k) - f^* \leq \varepsilon$.

Demonstração: Como $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ é compacto, f possui pelo menos um minimizador global x^*, o qual pertence a $\mathcal{L}(x_0)$. Pelo Lema 1.27, tem-se que

$$f(x_k) - f(x_{k+1}) \geq \frac{w}{L} \|\nabla f(x_k)\|^2, \tag{1.30}$$

para algum $w > 0$. Além disso, como f é convexa, segue do Lema 1.8 que

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

Em particular, para $x = x_k$ e $y = x^*$ tem-se que

$$f(x_k) - f^* \leq \nabla f(x_k)^T (x_k - x^*) \leq \|\nabla f(x_k)\| \, \|x_k - x^*\| .$$

Pelo fato de $\mathcal{L}(x_0)$ ser limitado, tem-se ainda $0 \leq \|x_k - x^*\| \leq D < \infty$, para todo k. Logo,

$$f(x_k) - f^* \leq D \, \|\nabla f(x_k)\| \quad \Rightarrow \quad \|\nabla f(x_k)\| \geq D^{-1} \left(f(x_k) - f^*\right). \tag{1.31}$$

Combinando (1.30) e (1.31), segue que

$$f(x_k) - f(x_{k+1}) \geq \frac{w}{LD^2} \left(f(x_k) - f^*\right)^2 .$$

Fazendo $\delta_k = f(x_k) - f^*$, obtém-se

$$\delta_k - \delta_{k+1} \geq \frac{w}{LD^2} \delta_k^2 .$$

Assim,

$$\frac{1}{\delta_{k+1}} - \frac{1}{\delta_k} = \frac{\delta_k - \delta_{k+1}}{\delta_k \delta_{k+1}} \geq \frac{\frac{w}{LD^2}\delta_k^2}{\delta_k \delta_{k+1}} > \frac{\frac{w}{LD^2}\delta_k^2}{\delta_k^2} = \frac{w}{LD^2}.$$

Dado $k > 0$, tem-se

$$\sum_{i=0}^{k-1} \frac{1}{\delta_{i+1}} - \frac{1}{\delta_i} \geq \sum_{i=0}^{k-1} \frac{w}{LD^2},$$

de onde,

$$\frac{1}{\delta_k} - \frac{1}{\delta_0} \geq k \frac{w}{LD^2}.$$

Logo,

$$\frac{1}{\delta_k} \geq \frac{1}{\delta_0} + k \frac{w}{LD^2} \geq k \frac{w}{LD^2}, \tag{1.32}$$

ou seja,

$$f(x_k) - f^* = \delta_k \leq \frac{LD^2}{kw}, \quad \forall \, k > 0.$$

Seja $\bar{k}$ o menor índice para o qual $\delta_{\bar{k}} = f(x_{\bar{k}}) - f^* < \varepsilon$. Então $\delta_{\bar{k}-1} \geq \varepsilon$ e, por (1.32),

$$\frac{1}{\varepsilon} \geq \frac{1}{\delta_{\bar{k}-1}} \geq (\bar{k} - 1) \frac{w}{LD^2}.$$

Logo,

$$\bar{k} \leq \frac{LD^2}{w\varepsilon} + 1 \leq \frac{LD^2 + w}{w\varepsilon}.$$

■

Definição 1.31. *Uma função $f : \mathbb{R}^n \to \mathbb{R}$ de classe $\mathcal{C}^1$ é dita fortemente convexa quando*

existe uma constante $\mu > 0$ tal que

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}\mu \|y-x\|^2, \tag{1.33}$$

para todos $x, y \in \mathbb{R}^n$. A constante μ é chamada de parâmetro de convexidade da função f.

Lema 1.32. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função fortemente convexa e x^* um minimizador global de f. Então,*

$$f(x) - f(x^*) \leq \frac{1}{2\mu} \|\nabla f(x)\|^2, \tag{1.34}$$

para todo $x \in \mathbb{R}^n$.

Demonstração: Dado $x \in \mathbb{R}^n$, considere

$$m_x(y) = f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|y-x\|^2.$$

Sejam x^* um minimizador global de f e $\bar{y}$ um minimizador global de $m_x(y)$. Então, segue de (1.33) que

$$m_x(\bar{y}) \leq m_x(x^*) = f(x) + \nabla f(x)^T(x^*-x) + \frac{1}{2}\mu\|x^*-x\|^2 \leq f(x^*). \tag{1.35}$$

Como $\bar{y}$ é minimizador global de $m_x(y)$, tem-se que

$$\nabla m_x(\bar{y}) = \nabla f(x) + \mu(\bar{y}-x) = 0,$$

ou seja,

$$\bar{y} = x - \frac{1}{\mu}\nabla f(x).$$

Logo,

$$\begin{aligned} m_x(\bar{y}) &= f(x) + \nabla f(x)^T\left(x - \frac{1}{\mu}\nabla f(x) - x\right) + \frac{\mu}{2}\left\|x - \frac{1}{\mu}\nabla f(x) - x\right\|^2 \\ &= f(x) - \frac{1}{\mu}\nabla f(x)^T\nabla f(x) + \frac{1}{2\mu}\|\nabla f(x)\|^2 \\ &= f(x) - \frac{1}{2\mu}\|\nabla f(x)\|^2. \end{aligned} \tag{1.36}$$

Combinando (1.35) e (1.36), conclui-se que

$$f(x^*) \geq f(x) - \frac{1}{2\mu}\|\nabla f(x)\|^2 \quad \Rightarrow \quad f(x) - f(x^*) \leq \frac{1}{2\mu}\|\nabla f(x)\|^2,$$

para todo $x \in \mathbb{R}^n$. ∎

Teorema 1.33. *Considere as mesmas hipóteses do Teorema 1.28. Suponha ainda que a*

função objetivo f é uma função fortemente convexa, com parâmetro $\mu > 0$. Então, tem-se que

$$f(x_k) - f^* \leq \left(1 - \frac{2\mu w}{L}\right)^k (f(x_0) - f^*), \quad \forall\, k. \tag{1.37}$$

Consequentemente, dado $0 < \varepsilon < 1$, o Método de Descida executa no máximo $\mathcal{O}(\log(\varepsilon^{-1}))$ iterações para gerar x_k tal que $f(x_k) - f^* \leq \varepsilon$.

Demonstração: Pelo Lema 1.27,

$$f(x_k) - f(x_{k+1}) \geq \frac{w}{L} \|\nabla f(x_k)\|^2. \tag{1.38}$$

Combinando (1.38) e (1.34), segue que

$$f(x_k) - f(x_{k+1}) \geq \frac{w}{L} 2\mu (f(x_k) - f^*). \tag{1.39}$$

Denotando $\delta_k = f(x_k) - f^*$, (1.39) pode ser escrita como

$$\delta_k - \delta_{k+1} \geq \frac{2\mu w}{L} \delta_k,$$

ou ainda

$$\delta_{k+1} \leq \left(1 - \frac{2\mu w}{L}\right) \delta_k.$$

Utilizando raciocínio indutivo, é fácil ver que

$$\delta_k \leq \left(1 - \frac{2\mu w}{L}\right)^k \delta_0, \quad \forall\, k,$$

o que é equivalente a (1.37).

Para a prova da segunda parte, basta notar que, se

$$\left(1 - \frac{2\mu w}{L}\right)^k \delta_0 \leq \varepsilon, \tag{1.40}$$

consequentemente tem-se $f(x_k) - f^* \leq \varepsilon$. Denote $q = 1 - \frac{2\mu w}{L}$. Observe que $0 < q < 1$. De fato, tem-se que $0 < \delta_{k+1} \leq q\delta_k \Rightarrow q > 0$ e $\frac{2\mu w}{L} > 0 \Rightarrow q < 1$. Impondo que (1.40)

seja verdadeira, tem-se que

$$\begin{aligned}
& \log\left(q^k\delta_0\right) \leq \log\left(\varepsilon\right) \\
\Leftrightarrow \quad & k\log q + \log\delta_0 \leq \log\varepsilon \\
\Leftrightarrow \quad & k\log q \leq \log\varepsilon - \log\delta_0 \\
\Leftrightarrow \quad & k \geq \frac{\log\varepsilon - \log\delta_0}{\log q} \\
\Leftrightarrow \quad & k \geq \frac{-\log\varepsilon}{|\log q|} + \frac{\log\delta_0}{|\log q|} \\
\Leftrightarrow \quad & k \geq \log(\varepsilon^{-1})\left(\frac{1+\frac{\log\delta_0}{\log\varepsilon^{-1}}}{|\log q|}\right).
\end{aligned}$$

Suponha, por exemplo, que $\varepsilon \leq \frac{1}{2} < 1$ [3]. Neste caso, tem-se

$$\log(\varepsilon^{-1})\left(\frac{1+\frac{\log\delta_0}{\log\varepsilon^{-1}}}{|\log q|}\right) \leq \log(\varepsilon^{-1})\left(\frac{1+\frac{|\log\delta_0|}{\log 2}}{|\log q|}\right).$$

Assim, se

$$k = \left\lceil \log(\varepsilon^{-1})\left(\frac{1+\frac{|\log\delta_0|}{\log 2}}{|\log q|}\right)\right\rceil$$

tem-se a garantia de que $f(x_k) - f^* < \varepsilon$. Portanto, a complexidade do Método de Descida para funções fortemente convexas é de ordem $\mathcal{O}\left(\log(\varepsilon^{-1})\right)$. ∎

1.3.2 Taxas de Convergência e Métodos quase-Newton

Diferentes métodos de otimização podem ser obtidos com diferentes escolhas para a matriz B_k. Por exemplo:

- Com $B_k = \nabla^2 f(x_k)^{-1}$, tem-se o Método de Newton, que foi o ponto de partida para dedução do Algoritmo 1.1;
- Com $B_k = I$ para todo k, tem-se o Método do Gradiente.

A escolha de $B_k = I$ se justifica pelo fato de que, neste caso, a direção de busca $d_k = -\nabla f(x_k)$ é a direção de maior decréscimo da função objetivo a partir do ponto x_k. De fato, se v é outra direção tal que $\|v\| = \|\nabla f(x)\|$, então

$$\frac{\partial f}{\partial d}(x) = \nabla f(x)^T d = -\left\|\nabla f(x)\right\|^2 = -\left\|v\right\|\left\|\nabla f(x)\right\| \leq -\left|\nabla f(x)^T v\right| \leq \nabla f(x)^T v = \frac{\partial f}{\partial v}(x),$$

ou seja, o decréscimo na direção d é mais acentuado do que na direção v.

[3] Poderia ser considerado $\varepsilon \leq \frac{1}{\alpha}$, com qualquer $\alpha > 1$.

Note que os limitantes de complexidade descritos na Subseção 1.3.1 não dependem da escolha de B_k. Assim, é justo questionar se existe alguma vantagem teórica do Método de Newton em relação ao Método do Gradiente. Afinal, a determinação de d_k no Método de Newton é muito mais complexa do que no Método do Gradiente, pois requer o cálculo da Hessiana $\nabla^2 f(x_k)$ e a resolução do sistema linear

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k). \tag{1.41}$$

É razoável se esperar que esse esforço computacional resulte em alguma melhora no desempenho do método em relação ao Método do Gradiente. Esta melhora realmente ocorre, e pode ser estabelecida teoricamente em termos das taxas de convergência de ambos os métodos.

Teorema 1.34. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ de classe $\mathcal{C}^2$. Suponha que $x^* \in \mathbb{R}^n$ seja um minimizador local de f, com $\nabla^2 f(x^*)$ definida positiva, e que a sequência x_k gerada pelo Método do Gradiente, com busca exata, converge para x^*. Então a sequência $(f(x_k))$ converge linearmente para $f(x^*)$ com taxa não superior a $\left(\dfrac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2$, onde λ_1 e λ_n são, respectivamente, o menor e o maior autovalor de $\nabla^2 f(x^*)$.*

Demonstração: Ver Seção 8.2 e 12.5 em Luenberger e Ye [19]. ∎

Teorema 1.35. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ de classe $\mathcal{C}^2$ com $\nabla^2 f$ Lipschitz. Suponha que $x^* \in \mathbb{R}^n$ seja um minimizador local de f, com $\nabla^2 f(x^*)$ definida positiva. Então, existe $\delta > 0$ tal que, se $x_0 \in B(x^*, \delta)$, o Método de Newton com $t_k = 1$ para todo k, gera uma sequência (x_k) que converge quadraticamente para x^*.*

Demonstração: Ver Teorema 5.10 em Ribeiro e Karas [25]. ∎

Apesar da convergência mais rápida do Método de Newton, o custo computacional para resolver o sistema linear (1.41) pode ser excessivamente alto, especialmente em problemas de grande porte (com $n \gg 1$). Isto motiva a busca por uma matriz B_k simétrica e definida positiva cuja construção não envolva o cálculo de $\nabla^2 f(x_k)$ ou a resolução de sistemas lineares, mas que ainda assim resulte em um método com convergência superlinear. Esse desejo de replicar a convergência rápida do Método de Newton a um custo similar ao do Método do Gradiente sugere que se busque, a cada iteração, uma aproximação $B_k \cong \nabla^2 f(x_k)^{-1}$ que possa ser construída usando-se apenas gradientes de f. Essa abordagem resulta nos chamados *Métodos quase-Newton*, os quais têm como ponto de partida o seguinte argumento: dados $x, y \in \mathbb{R}^n$ suficientemente próximos, pela fórmula de Taylor tem-se que

$$f(y) \cong f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x). \tag{1.42}$$

Então, definindo

$$m_x(y) = f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}(y-x)^T\nabla^2 f(x)(y-x),$$

a relação (1.42) sugere que

$$\nabla f(y) \cong \nabla m_x(y) = \nabla f(x) + \nabla^2 f(x)(y-x). \tag{1.43}$$

Fazendo $y = x_k$ e $x = x_{k+1}$ em (1.43), obtém-se

$$\nabla f(x_k) \cong \nabla f(x_{k+1}) + \nabla^2 f(x_{k+1})(x_k - x_{k+1}). \tag{1.44}$$

Por sua vez, (1.44) sugere que

$$(x_{k+1} - x_k) \cong \nabla^2 f(x_{k+1})^{-1} \left(\nabla f(x_{k+1}) - \nabla f(x_k)\right). \tag{1.45}$$

Denotando $\delta_k = x_{k+1} - x_k$ e $\gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, a relação (1.45) pode ser reescrita como

$$\nabla^2 f(x_{k+1})^{-1}\gamma_k \cong \delta_k. \tag{1.46}$$

Com base em (1.46), é natural que se busque B_{k+1} de modo que

$$B_{k+1}\gamma_k = \delta_k. \tag{1.47}$$

A equação (1.47) é conhecida como *Equação Secante*. Dada uma matriz B_k simétrica e positiva definida, o objetivo então é construir B_{k+1} que satisfaça (1.47). Uma maneira simples de se fazer isso consiste em perturbar B_k com uma matriz de posto 1, ou seja, tomar

$$B_{k+1} = B_k + uv^T,$$

onde $u, v \in \mathbb{R}^n$. Impondo a condição (1.47), segue-se que

$$\begin{aligned} B_{k+1}\gamma_k = \delta_k &\Leftrightarrow \left(B_k + uv^T\right)\gamma_k = \delta_k \\ &\Leftrightarrow uv^T\gamma_k = \delta_k - B_k\gamma_k \\ &\Leftrightarrow u = \frac{\delta_k - B_k\gamma_k}{v^T\gamma_k}, \end{aligned}$$

desde que se tenha $v^T\gamma_k \neq 0$. Assim,

$$B_{k+1} = B_k + \left(\frac{\delta_k - B_k\gamma_k}{v^T\gamma_k}\right) v^T.$$

Para garantir a simetria de B_{k+1}, uma escolha natural para v é $v = \delta_k - B_k\gamma_k$. O resultado

é

$$B_{k+1} = B_k + \frac{(\delta_k - B_k\gamma_k)(\delta_k - B_k\gamma_k)^T}{(\delta_k - B_k\gamma_k)^T \gamma_k}. \tag{1.48}$$

Por fim, para que se tenha B_{k+1} definida positiva, é suficiente que $(\delta_k - B_k\gamma_k)^T \gamma_k > 0$. A fórmula (1.48) é conhecida como SR1 (Symmetric rank 1), uma vez que tal B_{k+1} é obtida através de uma correção simétrica de posto 1 na matriz B_k.

Uma generalização dessa abordagem consiste em fazer uma perturbação de posto 2 da forma

$$B_{k+1} = B_k + \alpha uu^T + \beta vv^T,$$

com $\alpha, \beta \in \mathbb{R}$ e $u, v \in \mathbb{R}^n$. Impondo a equação secante (1.47), segue-se que

$$\begin{aligned} B_{k+1}\gamma_k = \delta_k &\Leftrightarrow \left(B_k + \alpha uu^T + \beta vv^T\right)\gamma_k = \delta_k \\ &\Leftrightarrow B_k\gamma_k + \alpha uu^T\gamma_k + \beta vv^T\gamma_k = \delta_k. \end{aligned} \tag{1.49}$$

Observe que se $\alpha uu^T\gamma_k = -B_k\gamma_k$ e $\beta vv^T\gamma_k = \delta_k$, então (1.49) será satisfeita. Para isso, basta tomar $u = B_k\gamma_k$, $\alpha = -\frac{1}{\gamma_k^T B_k \gamma_k}$, $v = \delta_k$ e $\beta = \frac{1}{\delta_k^T \gamma_k}$. Essas escolhas resultam em

$$B_{k+1} = B_k - \frac{B_k\gamma_k\gamma_k^T B_k}{\gamma_k^T B_k \gamma_k} + \frac{\delta_k\delta_k^T}{\delta_k^T\gamma_k} \tag{1.50}$$

A fórmula dada por (1.50) foi proposta independentemente por Davidon, Fletcher e Powell, e por isso é conhecida como fórmula DFP. Para que B_{k+1} em (1.50) seja definida positiva, é suficiente que se tenha $\delta_k^T\gamma_k > 0$.

Outra fórmula quase-Newton clássica para atualizar a matriz B_k é a fórmula BFGS (devida a Broyden, Fletcher, Goldfarb e Shanno). Para deduzir essa fórmula, toma-se como referência a equação secante para a própria matriz $\nabla^2 f(x_{k+1})$:

$$H_{k+1}\delta_k = \gamma_k. \tag{1.51}$$

Dada uma matriz B_k simétrica e definida positiva, denote $H_k = B_k^{-1}$. Então, de forma análoga à dedução da fórmula DFP, pode-se obter a partir de (1.51) a seguinte perturbação da matriz H_k:

$$H_{k+1} = H_k - \frac{H_k\delta_k\delta_k^T H_k}{\delta_k^T H_k \delta_k} + \frac{\gamma_k\gamma_k^T}{\delta_k^T\gamma_k}.$$

A fórmula BFGS resulta do cálculo de $B_{k+1} = H_{k+1}^{-1}$ pela fórmula de Sherman-Morrison. Como resultado, obtém-se

$$B_{k+1} = (H_{k+1})^{-1} = B_k + \left(1 + \frac{\gamma_k^T B_k \gamma_k}{\delta_k^T\gamma_k}\right)\frac{\delta_k\delta_k^T}{\delta_k^T\gamma_k} - \frac{\delta_k\gamma_k^T B_k + B_k\gamma_k\delta_k^T}{\delta_k^T\gamma_k}.$$

Também neste caso, para que B_{k+1} seja definida positiva é suficiente que se tenha $\delta_k^T\gamma_k >$

0.

Conforme discutido anteriormente, o interesse nas fórmulas quase-Newton reside na possibilidade de obter convergência superlinear a um custo comparável ao do Método do Gradiente. Uma condição suficiente para isso é dada pelo seguinte teorema:

Teorema 1.36 (Dennis-Moré)**.** *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função de classe $\mathcal{C}^2$. Suponha que x^* é um minimizador local de f, com $\nabla^2 f(x^*)$ definida positiva. Seja (x_k) uma sequência gerada pelo Algoritmo 1.1, com $x_k \to x^*$. Se*

$$\lim_{k\to\infty} \frac{\left\| \left(B_k^{-1} - \nabla^2 f(x^*)\right)(x_{k+1} - x_k) \right\|}{\|x_{k+1} - x_k\|} = 0 \quad e \quad t_k \to 1,$$

então (x_k) converge superlinearmente para x^*.

Demonstração: Ver Corolário 2.3 em Denis e Moré [10]. ∎

Sob certas condições, pode-se provar que as fórmulas quase-Newton descritas acima satisfazem as condições do Teorema 1.36. Para maiores detalhes, veja Denis e Moré [10, 11].

Com respeito às taxas de convergência, a Figura 1.3 resume as características dos métodos vistos até aqui.

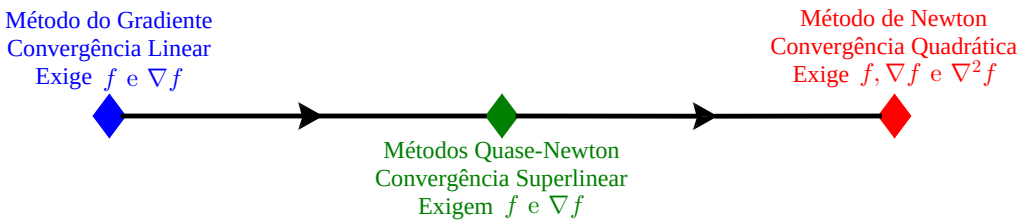


Figura 1.3: Espectro dos principais Métodos de Descida.

1.4 Método do Gradiente Acelerado

Os resultados de complexidade vistos na subseção 1.3.1 podem ser resumidos no seguinte quadro:

Funções não-convexas	$\mathcal{O}(\varepsilon^{-2})$
Funções convexas	$\mathcal{O}(\varepsilon^{-1})$
Funções fortemente convexas	$\mathcal{O}(\log(\varepsilon^{-1}))$.

Para funções convexas, o limitante de $\mathcal{O}(\varepsilon^{-1})$ iterações é subótimo. De fato, para essa classe de funções, a complexidade ótima para métodos de primeira ordem é de $\mathcal{O}(\varepsilon^{-\frac{1}{2}})$

iterações [23]. O Método do Gradiente Acelerado proposto por Nesterov em [24] consiste em uma modificação do Método do Gradiente que possui complexidade de pior caso ótima.

Dada uma função $f : \mathbb{R}^n \to \mathbb{R}$ diferenciável e convexa, com $\nabla f(x)$ L-Lipschitz, o Método do Gradiente Acelerado é dado pelo Algoritmo 1.2.

Algoritmo 1.2. *Método do Gradiente Acelerado de Nesterov*

Passo 0: Dado $x_0 \in \mathbb{R}^n$, *faça* $y_0 = x_0$; *Defina* $\lambda_0 = 0$ *e* $k = 0$.

Passo 1: Se $\nabla f(x_k) = 0$, *pare;*

Passo 2: Calcule $y_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$;

Passo 3: Calcule $\lambda_{k+1} = \dfrac{1+\sqrt{1+4\lambda_k^2}}{2}$ *(tem-se* $\lambda_{k+1}^2 - \lambda_{k+1} = \lambda_k^2$*);*

Calcule $\gamma_k = \dfrac{1-\lambda_k}{\lambda_{k+1}}$;

Passo 4: Defina $x_{k+1} = (1-\gamma_k)y_{k+1} + \gamma_k y_k$;

Passo 5: Defina $k = k+1$ *e volte ao Passo 1.*

Teorema 1.37. *([19], p.257) Seja f uma função convexa e diferenciável, com gradiente L-Lipschitz, e que admite um minimizador x^*. Então, o Algoritmo 1.2 gera uma sequência (y_k) tal que*

$$f(y_{k+1}) - f(x^*) \leq \frac{2L}{k^2}\|x_0 - x^*\|^2, \quad \forall k \geq 1.$$

Demonstração: Como f é convexa, por (1.3) tem-se que

$$f\left(x - \frac{1}{L}\nabla f(x)\right) - f(y) \leq f\left(x - \frac{1}{L}\nabla f(x)\right) - f(x) - \nabla f(x)^T(y-x). \tag{1.52}$$

Além disso, por (1.15) tem-se

$$\begin{aligned} f\left(x - \frac{1}{L}\nabla f(x)\right) - f(x) &\leq \frac{L}{2}\left\|x - \frac{1}{L}\nabla f(x) - x\right\|^2 + \nabla f(x)^T\left(x - \frac{1}{L}\nabla f(x) - x\right) \\ &= \frac{1}{2L}\|\nabla f(x)\|^2 - \frac{1}{L}\|\nabla f(x)\|^2 \\ &= -\frac{1}{2L}\|\nabla f(x)\|^2. \end{aligned} \tag{1.53}$$

Assim, por (1.52) e (1.53),

$$f\left(x - \frac{1}{L}\nabla f(x)\right) - f(y) \leq -\frac{1}{2L}\|\nabla f(x)\|^2 - \nabla f(x)^T(y-x). \tag{1.54}$$

Considerando $x = x_k$ e $y = y_k$ em (1.54) obtém-se

$$f(y_{k+1}) - f(y_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2 + \nabla f(x_k)^T(x_k - y_k). \tag{1.55}$$

Por outro lado, considerando $x = x_k$ e $y = x^*$ em (1.54) segue que

$$f(y_{k+1}) - f(x^*) \leq -\frac{1}{2L} \|\nabla f(x_k)\|^2 + \nabla f(x_k)^T (x_k - x^*). \tag{1.56}$$

Note que para $k \geq 1$, tem-se

$$\lambda_k = \frac{1 + \sqrt{1 + 4\lambda_{k-1}^2}}{2} \geq 1.$$

Denote $\delta_k = f(y_k) - f(x^*)$. Multiplicando ambos os membros de (1.55) por $(\lambda_k - 1)$ e adicionando o resultado, membro a membro, à (1.56), obtém-se

$$\begin{aligned}
(\lambda_k - 1)(\delta_{k+1} - \delta_k) + \delta_{k+1} &\leq (\lambda_k - 1)\nabla f(x_k)^T (x_k - y_k) + \nabla f(x_k)^T (x_k - x^*) \\
&\quad - \frac{1}{2L} \|\nabla f(x_k)\|^2 - (\lambda_k - 1)\frac{1}{2L} \|\nabla f(x_k)\|^2 \\
&= \nabla f(x_k)^T \left((\lambda_k - 1)(x_k - y_k) + (x_k - x^*)\right) - \frac{\lambda_k}{2L} \|\nabla f(x_k)\|^2 \\
&= -L(y_{k+1} - x_k)^T (\lambda_k x_k - (\lambda_k - 1)y_k - x^*) - \frac{\lambda_k}{2L} \|L(y_{k+1} - x_k)\|^2
\end{aligned}$$

que pode ser reescrita como

$$\lambda_k \delta_{k+1} - (\lambda_k - 1)\delta_k = -\frac{L}{2} \left[2(y_{k+1} - x_k)^T (\lambda_k x_k - (\lambda_k - 1)y_k - x^*) + \lambda_k \|y_{k+1} - x_k\|^2\right]. \tag{1.57}$$

Multiplicando ambos os membros de (1.57) por λ_k, e utilizando o fato de que $\lambda_k^2 - \lambda_k = \lambda_{k-1}^2$, obtém-se

$$\lambda_k^2 \delta_{k+1} - \lambda_{k-1}^2 \delta_k \leq -\frac{L}{2} \left[2\lambda_k (y_{k+1} - x_k)^T (\lambda_k x_k - (\lambda_k - 1)y_k - x^*) + \|\lambda_k (y_{k+1} - x_k)\|^2\right].$$

Observe que

$$\begin{aligned}
\|\lambda_k y_{k+1} - (\lambda_k - 1)y_k - x^*\|^2 &= \|\lambda_k (y_{k+1} - x_k) + \lambda_k x_k - (\lambda_k - 1)y_k - x^*\|^2 \\
&= \|\lambda_k (y_{k+1} - x_k)\|^2 + \|\lambda_k x_k - (\lambda_k - 1)y_k - x^*\|^2 \\
&\quad + 2\lambda_k (y_{k+1} - x_k)^T \left[\lambda_k x_k - (\lambda_k - 1)y_k - x^*\right],
\end{aligned}$$

e portanto

$$\lambda_k^2 \delta_{k+1} - \lambda_{k-1}^2 \delta_k \leq -\frac{L}{2} \left(\|\lambda_k y_{k+1} - (\lambda_k - 1)y_k - x^*\|^2 - \|\lambda_k x_k - (\lambda_k - 1)y_k - x^*\|^2\right). \tag{1.58}$$

Além disso, segue da definição de x_k e de γ_k que

$$\begin{aligned} & x_k = (1-\gamma_{k-1})y_k + \gamma_{k-1}y_{k-1} = y_k + \gamma_{k-1}(y_{k-1} - y_k) \\ \Rightarrow \quad & \lambda_k x_k = \lambda_k y_k + \lambda_k\gamma_{k-1}(y_{k-1} - y_k) = \lambda_k y_k + (1-\lambda_{k-1})(y_{k-1} - y_k) \\ \Rightarrow \quad & \lambda_k x_k - (\lambda_k - 1)y_k = \lambda_k y_k + (1-\lambda_{k-1})(y_{k-1} - y_k) - (\lambda_k - 1)y_k \\ \Rightarrow \quad & \lambda_k x_k - (\lambda_k - 1)y_k = \lambda_{k-1} y_k - (\lambda_{k-1} - 1)y_{k-1}. \end{aligned} \tag{1.59}$$

Substituindo (1.59) em (1.58), tem-se

$$\lambda_k^2\delta_{k+1} - \lambda_{k-1}^2\delta_k \leq -\frac{L}{2}\left(\|\lambda_k y_{k+1} - (\lambda_k - 1)y_k - x^*\|^2 - \|\lambda_{k-1}y_k - (\lambda_{k-1} - 1)y_{k-1} - x^*\|^2\right). \tag{1.60}$$

Denotando $u_k = \lambda_k y_{k+1} - (\lambda_k - 1)y_k - x^*$, a desigualdade (1.60) assume a forma

$$\lambda_k^2\delta_{k+1} - \lambda_{k-1}^2\delta_k \leq -\frac{L}{2}\left(\|u_k\|^2 - \|u_{k-1}\|^2\right) = \frac{L}{2}\left(\|u_{k-1}\|^2 - \|u_k\|^2\right). \tag{1.61}$$

A partir de (1.61), obtém-se

$$\begin{aligned} & \sum_{i=1}^{k}\left(\lambda_i^2\delta_{i+1} - \lambda_{i-1}^2\delta_i\right) \leq \sum_{i=1}^{k}\frac{L}{2}\left(\|u_{i-1}\|^2 - \|u_i\|^2\right) \\ \Rightarrow \quad & \lambda_k^2\delta_{k+1} - \lambda_0^2\delta_1 \leq \frac{L}{2}\left(\|u_0\|^2 - \|u_k\|^2\right) \leq \frac{L}{2}\|u_0\|^2, \end{aligned}$$

ou seja,

$$\delta_{k+1} \leq \frac{L\|u_0\|^2}{2\lambda_k^2},$$

uma vez que $\lambda_0 = 0$. Por fim, tem-se que

- $\delta_{k+1} = f(y_{k+1}) - f(x^*)$;
- $u_0 = \lambda_0 y_1 - (\lambda_0 - 1)y_0 - x^* = y_0 - x^* = x_0 - x^*$, pois $\lambda_0 = 0$ e $y_0 = x_0$;
- $\lambda_k \geq \frac{k}{2}$, $\forall\ k \geq 1$. Utilizando indução matemática: para $k = 1$ tem-se $\lambda_1 = 1 \geq \frac{1}{2}$, satisfazendo a desigualdade. Supondo válida para k, obtém-se:

$$\lambda_{k+1} = \frac{1 + \sqrt{1 + 4\lambda_k^2}}{2} \geq \frac{1 + \sqrt{4\left(\frac{k}{2}\right)^2}}{2} = \frac{1+k}{2}.$$

Logo,

$$f(y_{k+1}) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{k^2}, \quad \forall\ k \geq 1.$$

■

Corolário 1.38. *A ordem de complexidade do método dado pelo Algoritmo 1.2 é de* $\mathcal{O}(\varepsilon^{-\frac{1}{2}})$.

Demonstração: Pelo teorema anterior, tem-se

$$f(y_{k+1}) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{k^2}, \quad \forall\, k \geq 1.$$

Para que $f(y_{k+1}) - f(x^*) \leq \varepsilon$, é suficiente que

$$\frac{2L\|x_0 - x^*\|^2}{k^2} \leq \varepsilon,$$

ou seja,

$$k \geq \left(\frac{2L\|x_0 - x^*\|^2}{\varepsilon}\right)^{\frac{1}{2}} = \left(\sqrt{2L}\,\|x_0 - x^*\|\right)\varepsilon^{-\frac{1}{2}}.$$

Portanto, tomando

$$k = \left\lceil \left(\sqrt{2L}\,\|x_0 - x^*\|\right)\varepsilon^{-\frac{1}{2}} \right\rceil$$

tem-se que $f(y_{k+1}) - f(x^*) \leq \varepsilon$. ∎

Capítulo 2

Modelos de Aprendizagem de Máquina

Nos últimos anos, com o barateamento de computadores, smartphones, câmeras e sensores, mais pessoas e empresas passaram a ter acesso a essas tecnologias. A consequência disso foi um crescimento exponencial no volume de dados gerados a partir desses dispositivos, e cujo armazenamento passou a ser viável. A disponibilidade desses dados tem impulsionado uma demanda significativa por métodos computacionais capazes de extrair informações relevantes dos mesmos. Nesse contexto, a Aprendizagem de Máquina pode ser definida genericamente como a área multidisciplinar que se ocupa com o desenvolvimento, análise e aplicação de métodos para a detecção automática de padrões em conjuntos de dados. Esses conjuntos podem assumir diversas formas, tais como: dados bancários, preferências de usuários durante a navegação na Internet, dados sobre pacientes e suas doenças, histórico de compras de clientes em supermercados, e séries temporais sobre fenômenos diversos. Seja qual for o tipo de dado, o objetivo final da aprendizagem automática de padrões é a mineração de informações úteis que possam orientar a tomada de decisões sobre o problema em estudo. No caso de problemas envolvendo séries temporais, a obtenção de previsões costuma ter um papel fundamental.

Este capítulo tem como foco a descrição dos quatro modelos matemáticos de aprendizagem de máquina utilizados nesta dissertação para a previsão dos movimentos do Ibovespa, a saber: Regressão Linear, Regressão Logística, Máquinas de Vetor Suporte e Redes Neurais Artificias. As principais referências para este capítulo são Bishop [2], Murphy [22], Goodfellow et al [13], Ng [1] e Haykin [14].

2.1 Regressão Linear

Considere um conjunto de dados $\Omega = \left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(s)}, y^{(s)})\right\} \subset \mathbb{R}^2$. O problema de Regressão Simples consiste em determinar uma função $m : \mathbb{R} \to \mathbb{R}$ tal que a distância

entre os pontos de Ω e o gráfico de m seja a menor possível. O modelo de Regressão Linear Simples é o caso trivial no qual supõe-se que m é uma função afim, isto é,

$$m(x) = \theta_0 + \theta_1 x,$$

onde $\theta_0, \theta_1 \in \mathbb{R}$. Denotando $m(x) = m_\theta(x)$ para explicitar a dependência de m em relação ao parâmetro $\theta = (\theta_0, \theta_1)$, o problema de Regressão Linear Simples se traduz no problema de otimização sem restrições

$$\min_{\theta \in \mathbb{R}^2} f(\theta) = \sum_{i=1}^{s} \left(y^{(i)} - m_\theta(x^{(i)})\right)^2.$$

Geometricamente, isto corresponde à busca pela reta que melhor se ajusta aos pontos dados por Ω. Apesar da sua simplicidade, o modelo de Regressão Linear pode ser bastante útil para se fazer previsões. Por exemplo, suponha que $x^{(i)}$ é a distância percorrida por um automóvel (em quilômetros) e que $y^{(i)}$ é a quantidade de combustível (em litros) gasta nesse percurso. Usando Regressão Linear, pode-se determinar a reta que melhor se ajusta aos dados $(x^{(i)}, y^{(i)})$, e então utilizá-la para prever a quantidade de combustível necessária para percorrer $\tilde{x}$ quilômetros, com $\tilde{x} \notin \Omega$. Esta situação está ilustrada na Figura 2.1, onde os asteriscos indicam os dados conhecidos e a reta é aquela obtida por Regressão Linear[1].

No entanto, o uso de apenas uma variável explicativa (ou variável independente) geralmente não é suficiente para descrever certas situações. O preço de uma casa, por exemplo, é influenciado por diversos fatores, como área total, quantidade de quartos e de banheiros, localização, conservação, etc. Mesmo no exemplo anterior, seria interessante acrescentar outras variáveis, como a potência do automóvel (carros mais potentes costumam consumir mais combustível). Nestes casos tem-se a chamada Regressão Linear Múltipla, que ao invés de uma reta procura por um hiperplano que se ajuste ao conjunto de dados.

A formulação do problema é feita de maneira semelhante. Considere um conjunto de dados $\Omega = \left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(s)}, y^{(s)})\right\}$, com $x^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)}) \in \mathbb{R}^n$ e $y^{(i)} \in \mathbb{R}$, para $i = 1, \ldots, s$. O caso $n = 1$ corresponde à Regressão Linear Simples. O ideal seria encontrar uma função afim

$$m_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

tal que

$$m_\theta(x^{(i)}) = y^{(i)}, \quad i = 1, \ldots, s. \tag{2.1}$$

Observe que, acrescentando $x_0^{(i)} = 1$ a cada amostra $x^{(i)}$ do conjunto Ω, é possível rees-

[1]Os dados deste exemplo foram retirados do site `http://www.portalaction.com.br/analise-de-regressao/exercicios`, acessado em 2 de março de 2017.

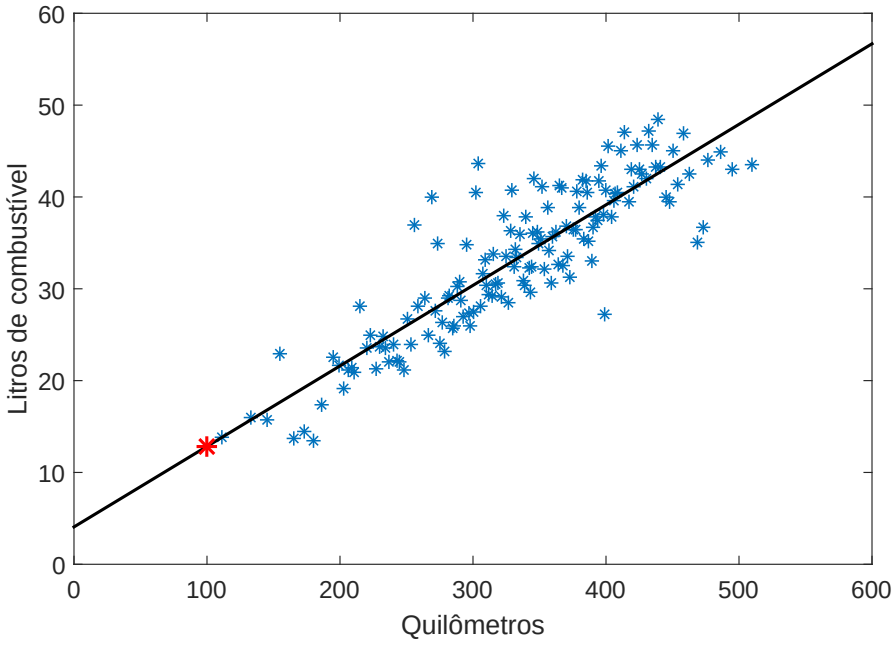


Figura 2.1: Exemplo de Regressão Linear relacionando a distância percorrida por um automóvel com o consumo de combustível.
Neste exemplo, o modelo estima um gasto de 13 litros de combustível para um trajeto de 100 Km.

crever $m_\theta(x^{(i)}) = \theta^T x^{(i)}$. Matricialmente, (2.1) assume a forma

$$\underbrace{\begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(s)} & \dots & x_n^{(s)} \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(s)} \end{bmatrix}}_{y}.$$

Em geral $s > n$, ou seja, o sistema linear gerado é sobredeterminado, podendo não existir θ tal que $\|X\theta - y\| = 0$. Isto motiva a determinação de θ pela resolução do problema de mínimos quadrados linear

$$\min_{\theta \in \mathbb{R}^{n+1}} f(\theta) = \|X\theta - y\|^2 = \sum_{i=1}^{s} \left(m_\theta(x^{(i)}) - y^{(i)}\right)^2. \tag{2.2}$$

Geometricamente, dado um conjunto de dados $\Omega \subset \mathbb{R}^{n+1}$, busca-se o hiperplano (ou reta, no caso em que $n = 1$) que melhor se ajusta a tal conjunto, no sentido de minimizar a soma das distâncias (ao quadrado) dos pontos ao hiperplano. Na Figura 2.2, ilustramos um exemplo para o caso em que $n = 2$.

Para resolver o problema (2.2), pode-se fazer uso de decomposição matricial, como decomposição QR, SVD ou fatoração de Cholesky.

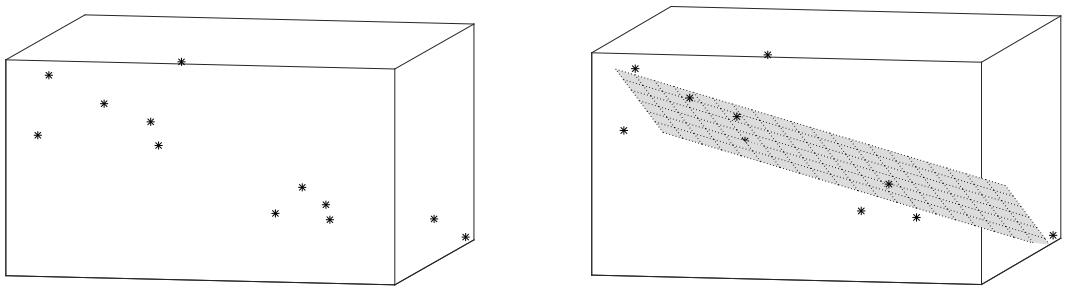


Figura 2.2: Conjunto de dados e o respectivo plano que melhor ajusta tais dados.

Note que

$$f(\theta) = \|X\theta - y\|^2 = \theta^T X^T X\theta - 2(X^T y)^T\theta + y^T y.$$

Logo,

$$\nabla f(\theta) = 2X^T X\theta - 2X^T y,$$

e

$$\nabla^2 f(\theta) = 2X^T X.$$

Como $\nabla^2 f(\theta)$ é semidefinida positiva, segue-se que f é convexa. Assim, θ é um minimizador global de f se, e somente se, $\nabla f(\theta) = 0$, o que é equivalente a

$$X^T X\theta = X^T y. \tag{2.3}$$

A equação linear (2.3) é conhecida como *Equação Normal*. No caso em que a matriz X tem posto coluna completo, tem-se que $X^T X$, além de ser simétrica, também é definida positiva. Consequentemente, a equação (2.3) possui uma única solução, e pode ser resolvida usando a Fatoração de Cholesky.

Teorema 2.1 (Fatoração de Cholesky)**.** *Seja $A \in \mathbb{R}^{n\times n}$ uma matriz simétrica e definida positiva. Então exite uma matriz L triangular inferior tal que*

$$A = LL^T.$$

Demonstração: Ver Teorema 4.2.7 em Golub e Loan [12]. ∎

A resolução da equação normal a partir da fatoração de Cholesky segue os seguintes passos:

- Encontrar a decomposição Cholesky de $X^T X$, ou seja, calcular L tal que $X^T X = LL^T$;
- Resolver o sistema triangular inferior $Lw = X^T y$;
- Resolver o sistema triangular superior $L^T\theta = w$.

No caso em que X não tem posto completo, X^TX pode ser singular, inviabilizando a determinação de θ pela Equação Normal. É possível torná-la positiva definida acrescentando-lhe uma matriz λI, com $\lambda > 0$ uma constante apropriada. Neste caso, tem-se

$$z^T(X^TX + \lambda I)z = z^TX^TXz + \lambda z^Tz > 0, \; \forall \, z \neq 0.$$

Este acréscimo se dá de forma automática se o termo de regularização $\lambda \|\theta\|^2$ for acrescentado ao problema original, resultando no problema

$$\min_{\theta \in \mathbb{R}^n} \quad f(\theta) = \|X\theta - y\|^2 + \lambda \|\theta\|^2 . \tag{2.4}$$

De fato, calculando o gradiente de f em (2.4) obtém-se

$$\nabla f(\theta) = 2X^TX\theta - 2X^Ty + 2\lambda\theta = 2(X^TX + \lambda I)\theta - 2X^Ty.$$

Assim, a equação normal a ser resolvida neste caso é

$$(X^TX + \lambda I)\theta = X^Ty.$$

Teorema 2.2 (Decomposição QR)**.** *Se $A \in \mathbb{R}^{m\times n}$, então existe uma matriz ortogonal $Q \in \mathbb{R}^{m\times m}$ e uma matriz triangular superior $R \in \mathbb{R}^{m\times n}$ tal que $A = QR$.*

Demonstração: Ver Teorema 5.2.1 em Golub e Loan [12]. ∎

Teorema 2.3. *Seja $A \in \mathbb{R}^{m\times n}$ com posto coluna completo. A fatoração QR reduzida $A = \tilde{Q}\tilde{R}$ é única, onde $\tilde{Q} \in \mathbb{R}^{m\times n}$ tem colunas ortonormais e $\tilde{R} \in \mathbb{R}^{n\times n}$ é triangular superior com entradas da diagonal positivas.*

Demonstração: Ver Teorema 5.2.3 em Golub e Loan [12]. ∎

Utilizando Decomposição QR, se X tem posto coluna completo a Equação Normal (2.3) pode ser escrita como

$$X^TX\theta = X^Ty \;\Rightarrow\; \tilde{R}^T\tilde{Q}^T\tilde{Q}\tilde{R}\theta = \tilde{R}^T\tilde{Q}^Ty \;\Rightarrow\; \tilde{R}^T\tilde{R}\theta = \tilde{R}^T\tilde{Q}^Ty \;\Rightarrow\; \tilde{R}\theta = \tilde{Q}^Ty,$$

e uma solução θ do problema de mínimos quadrados pode ser obtida resolvendo-se o sistema triangular superior $\tilde{R}\theta = z$, onde $z = \tilde{Q}^Ty$.

Teorema 2.4 (Decomposição SVD)**.** *Se $A \in \mathbb{R}^{m\times n}$, existem matrizes ortogonais $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{n\times n}$ tais que*

$$U^TAV = \Sigma = diag(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n}, \qquad p = \min\{m, n\},$$

onde $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

Demonstração: Ver Teorema 2.4.1 em Golub e Loan [12]. ∎

Se

$$X = U\Sigma V^T = \sum_{i=1}^{n} \sigma_i u_i v_i^T$$

é a SVD de uma matriz de posto cheio $X \in \mathbb{R}^{s\times n}$, com $s > n$, então

$$\|X\theta - y\|^2 = \left\|(U^T XV)(V^T\theta) - U^T y\right\|^2 = \sum_{i=1}^{n} \left(\sigma_i z_i - (u_i^T y)\right)^2 + \sum_{i=n+1}^{s} (u_i^T y)^2, \qquad (2.5)$$

onde $z = V^T\theta$. (2.5) é minimizada por $z_i = \frac{u_i^T y}{\sigma_i}$, $i = 1, \ldots, n$. Assim, uma solução para o problema de mínimos quadrados utilizando SVD seria

$$\theta = \sum_{i=1}^{n} \frac{u_i^T y}{\sigma_i} v_i.$$

Ao se resolver o problema de otimização (2.2) (ou (2.4)), obtém-se um parâmetro θ. Com esse parâmetro, dada uma nova entrada $x \notin \Omega$, basta calcular $m_\theta(x) = \theta^T x$ para obter uma previsão ou aproximação da saída y correspondente.

2.2 Regressão Logística

Considere um conjunto de dados

$$\Omega = \left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(s)}, y^{(s)})\right\},$$

onde $x^{(i)} = (x_0^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)}) \in \mathbb{R}^{n+1}$, com $x_0^{(i)} = 1$, e $y^{(i)} \in \{0, 1\}$, para $i = 1, \ldots, s$. Quando $y^{(i)} = 1$, diz-se que $x^{(i)}$ corresponde à classe positiva, e quando $y^{(i)} = 0$ diz-se que $x^{(i)}$ corresponde à classe negativa. Um classificador perfeito para o conjunto Ω seria uma função $c : \mathbb{R}^{n+1} \to \{0, 1\}$ tal que

$$c(x^{(i)}) = y^{(i)}, \quad i = 1, \ldots, s.$$

Por simplicidade, suponha que essa função c é especificada por um parâmetro $\theta = (\theta_0, \theta_1, \ldots, \theta_n) \in \mathbb{R}^{n+1}$ segundo a regra

$$c(x) = \begin{cases} 1 & \text{se } \theta^T x \geq 0, \\ 0 & \text{se } \theta^T x < 0. \end{cases} \qquad (2.6)$$

Denote as classes positiva e negativa, respectivamente, por

$$P = \left\{(x_1^{(i)}, \ldots, x_n^{(i)}) \,|\, y^{(i)} = 1,\ i = 1, \ldots, s\right\}$$

e

$$N = \left\{ (x_1^{(i)}, \ldots, x_n^{(i)}) \,|\, y^{(i)} = 0,\ i = 1, \ldots, s \right\}.$$

Neste caso, geometricamente, a classificação dos pontos de Ω corresponde à separação dos conjuntos P e N em $\mathbb{R}^n$ por um hiperplano $\theta^T x = 0$. A Figura 2.3 mostra um exemplo de classificação desse tipo envolvendo duas espécies de flor Íris: a Íris Setosa ($y = 0$) e a Íris Versicolor ($y = 1$). Cada amostra de flor é especificada por um ponto (x_1, x_2), onde x_1 é o comprimento e x_2 é a largura das pétalas da flor. Os triângulos representam amostras de Íris Versicolor e os quadrados representam amostras de Íris Setosa. Conhecido o hiperplano separador, é possível então identificar a espécie de uma nova amostra de flor com base no lado do hiperplano em que a mesma se coloca. Para este exemplo, a nova amostra seria classificada como sendo pertencente à espécie Íris Versicolor.

Evidentemente, o ponto chave desse processo de classificação é a determinação de um hiperplano separador $\theta^T x = 0$, ou seja, a busca pelo parâmetro $\theta \in \mathbb{R}^n$. Além disso, é fundamental que a determinação desse parâmetro possa ser feita mesmo quando a classificação perfeita não seja possível. O modelo de Regressão Logística fornece uma maneira para se calcular tal hiperplano considerando a probabilidade de um ponto pertencer à classe positiva. Para isto, utiliza-se

$$m_\theta(x) = g(\theta^T x), \text{ com } g(z) = \frac{1}{1 + e^{-z}}. \tag{2.7}$$

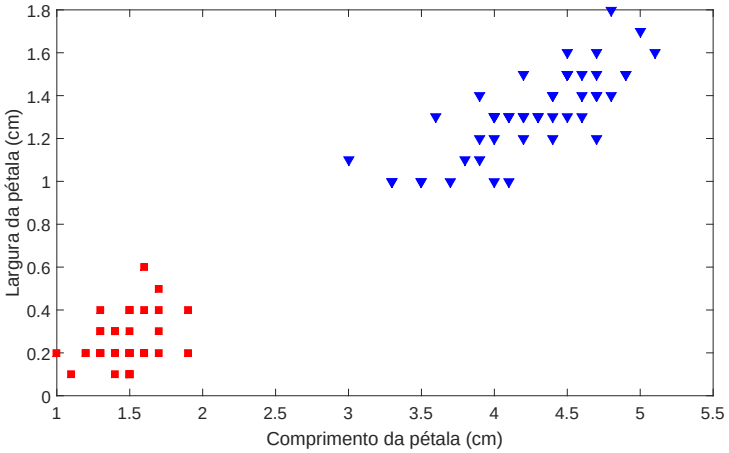
O gráfico da função g, conhecida como função logística, pode ser visto na Figura 2.4. Note que g é uma função crescente tal que:

- $g(z) \in [0, 1]$, para todo $z \in \mathbb{R}$;
- $\lim_{z \to -\infty} g(z) = 0$;
- $\lim_{z \to +\infty} g(z) = 1$.

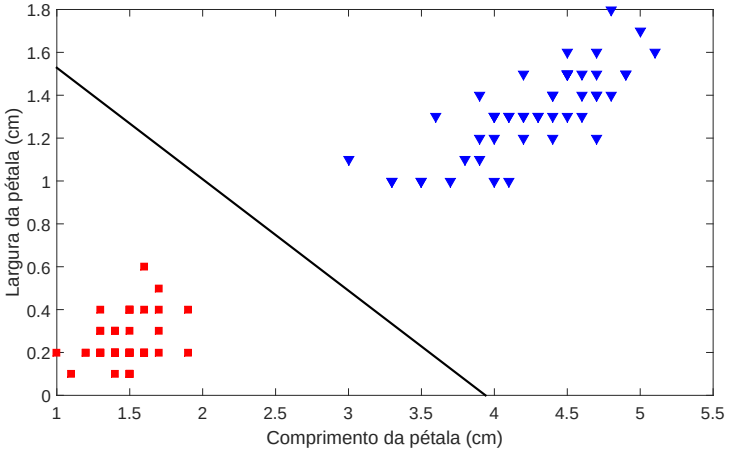
Deste modo, o valor $m_\theta(x)$ pode ser interpretado como a probabilidade do ponto x pertencer à classe positiva. A classificação, então, ocorre da seguinte maneira: se $m_\theta(x) \geq 0.5$, diz-se que x pertence à classe positiva, caso contrário diz-se que x pertence à classe negativa. Note que $m_\theta(x) \geq 0.5$ se, e somente se, $\theta^T x \geq 0$. Assim, a partir de $m_\theta(x)$ tem-se um classificador da forma (2.6). A determinação de θ no modelo de Regressão Logística poderia ser feita como na Regressão Linear, pela minimização da função

$$f(\theta) = \sum_{i=1}^{s} (m_\theta(x^{(i)}) - y^{(i)})^2, \tag{2.8}$$

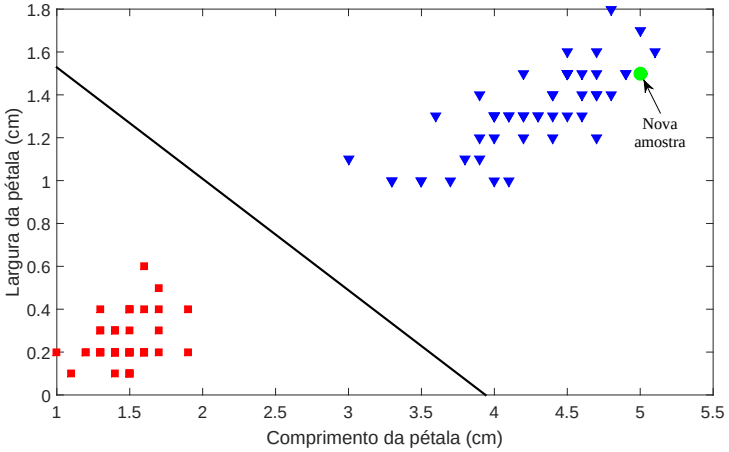
agora com $m_\theta(x)$ dada por (2.7). Isto faria sentido para o problema, uma vez que busca-se por um parâmetro θ para o qual se tenha



(a) Conjunto de dados referentes à flor Íris.



(b) Hiperplano separador.



(c) Nova amostra.

Figura 2.3: Exemplo de Regressão Logística.

- $m_\theta(x^{(i)}) \approx 1$ se $y^{(i)} = 1$, e
- $m_\theta(x^{(i)}) \approx 0$ se $y^{(i)} = 0$.

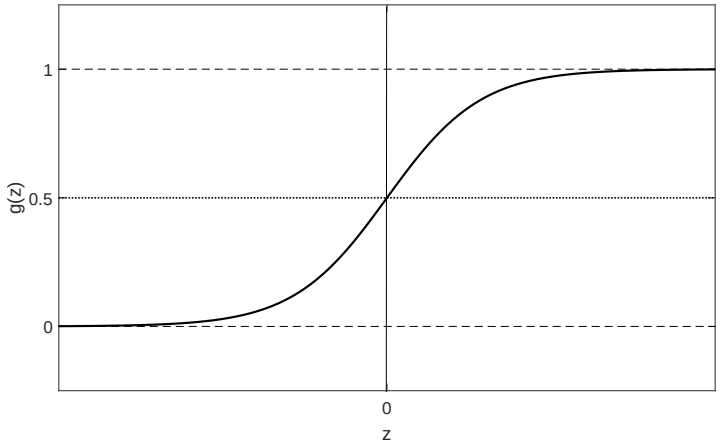


Figura 2.4: Gráfico da Função Logística.

No entanto, diferentemente da Regressão Linear, a função objetivo f dada em (2.8) não é necessariamente convexa, o que torna a sua minimização global extremamente difícil. Para contornar esse problema, a determinação do parâmetro θ no modelo de Regressão Logística é feita minimizando-se a função

$$f(\theta) = -\sum_{i=1}^{s} \left[y^{(i)} \log(m_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - m_\theta(x^{(i)})) \right]. \tag{2.9}$$

Tal função f é apropriada para se obter θ coerente com a tarefa de classificação, visto que:

- Se $y^{(i)} = 1$, $-\log(m_\theta(x^{(i)}))$ tende para 0 quando $m_\theta(x^{(i)})$ se aproxima de 1, e tende para $+\infty$ quando $m_\theta(x^{(i)})$ se aproxima de 0.
- Se $y^{(i)} = 0$, $-\log(1 - m_\theta(x^{(i)}))$ tende para 0 quando $m_\theta(x^{(i)})$ se aproxima de 0, e tende para $+\infty$ quando $m_\theta(x^{(i)})$ se aproxima de 1.

Mas a principal vantagem da função objetivo em (2.9) comparada com (2.8) decorre do resultado apresentado no Teorema 2.5.

Teorema 2.5. *A função f dada por* (2.9) *é uma função convexa.*

Demonstração: Conforme o Teorema 1.11, basta provar que $\nabla^2 f(\theta)$ é semidefinida positiva, ou seja, que $\nabla^2 f(\theta) \geq 0$, para todo θ. Calculando a derivada de f, tem-se que

$$\frac{\partial f}{\partial \theta_j}(\theta) = \sum_{i=1}^{s} \left(m_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}. \tag{2.10}$$

De fato,

$$\frac{\partial}{\partial \theta_j} \left(\log(1 + e^{-\theta^T x}) \right) = \frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}}(-x_j) = -\frac{1}{1 + e^{\theta^T x}} x_j = -m_\theta(-x) x_j$$

e

$$\frac{\partial}{\partial \theta_j}\left(\log(1+e^{\theta^T x})\right) = \frac{e^{\theta^T x}}{1+e^{\theta^T x}}x_j = \frac{1}{1+e^{-\theta^T x}}x_j = m_\theta(x)x_j.$$

Observe que

1. $\log\left(m_\theta(x)\right) = \log\left(\frac{1}{1+e^{-\theta^T x}}\right) = \log(1) - \log(1+e^{-\theta^T x}) = -\log(1+e^{-\theta^T x})$;
2. $\log\left(1-m_\theta(x)\right) = \log\left(\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}\right) = \log\left(\frac{1}{1+e^{\theta^T x}}\right) = -\log(1+e^{\theta^T x})$;
3. $1 - m_\theta(x) = 1 - \frac{1}{1+e^{-\theta^T x}} = \frac{e^{-\theta^T x}}{1+e^{-\theta^T x}} = \frac{e^{-\theta^T x}}{e^{-\theta^T x}(e^{\theta^T x}+1)} = \frac{1}{1+e^{\theta^T x}} = m_\theta(-x)$.

Assim,

$$\begin{aligned}
\frac{\partial f}{\partial \theta_j}(\theta) &= \frac{\partial}{\partial \theta_j}\left(-\sum_{i=1}^{s}\left[y^{(i)}\log\left(m_\theta(x^{(i)})\right) + (1-y^{(i)})\log\left(1-m_\theta(x^{(i)})\right)\right]\right) \\
&= \frac{\partial}{\partial \theta_j}\left(-\sum_{i=1}^{s}\left[-y^{(i)}\log\left(1+e^{-\theta^T x^{(i)}}\right) - (1-y^{(i)})\log\left(\log(1+e^{\theta^T x^{(i)}})\right)\right]\right) \\
&= \sum_{i=1}^{s}\left[y^{(i)}\left(-m_\theta(-x^{(i)})\right)x_j^{(i)} + (1-y^{(i)})m_\theta(x^{(i)})x_j^{(i)}\right] \\
&= \sum_{i=1}^{s}\left[-y^{(i)}\left(1-m_\theta(x^{(i)})\right)x_j^{(i)} + (1-y^{(i)})m_\theta(x^{(i)})x_j^{(i)}\right] \\
&= \sum_{i=1}^{s}\left(-y^{(i)} + y^{(i)}m_\theta(x^{(i)}) + m_\theta(x^{(i)}) - y^{(i)}m_\theta(x^{(i)})\right)x_j^{(i)} \\
&= \sum_{i=1}^{s}\left(m_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)}.
\end{aligned}$$

Logo, considerando $M = \begin{bmatrix} m_\theta(x^{(1)}) & \dots & m_\theta(x^{(s)}) \end{bmatrix}^T$, a expressão (2.10) pode ser reescrita como

$$\frac{\partial f}{\partial \theta_j}(\theta) = \begin{bmatrix} x_j^{(1)} & \dots & x_j^{(s)} \end{bmatrix}(M - Y).$$

Consequentemente

$$\nabla f(\theta) = X^T(M-Y),$$

com

$$X = \begin{bmatrix} x_0^{(1)} & \dots & x_n^{(1)} \\ \vdots & \ddots & \vdots \\ x_0^{(s)} & \dots & x_n^{(s)} \end{bmatrix} \text{ e } Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(s)} \end{bmatrix} \tag{2.11}$$

Além disso, tem-se que

$$\frac{\partial^2 f}{\partial \theta_k \partial \theta_j}(\theta) = \sum_{i=1}^{s} x_j^{(i)}x_k^{(i)}m_\theta(x^{(i)})\left(1-m_\theta(x^{(i)})\right).$$

De fato,

$$\begin{aligned}
\frac{\partial^2 f}{\partial\theta_k\partial\theta_j}(\theta) &= \frac{\partial f}{\partial\theta_k}\left(\sum_{i=1}^{s}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}}-y^{(i)}\right)x_j^{(i)}\right)\\
&= \sum_{i=1}^{s}\frac{\partial f}{\partial\theta_k}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}}x_j^{(i)}\right)\\
&= \sum_{i=1}^{s}\frac{-x_j^{(i)}(-x_k^{(i)})e^{-\theta^T x^{(i)}}}{\left(1+e^{-\theta^T x^{(i)}}\right)^2}\\
&= \sum_{i=1}^{s}x_j^{(i)}x_k^{(i)}e^{-\theta^T x^{(i)}}\left(m_\theta(x^{(i)})\right)^2\\
&= \sum_{i=1}^{s}x_j^{(i)}x_k^{(i)}m_\theta(x^{(i)})\left(1-m_\theta(x^{(i)})\right).
\end{aligned}$$

Denotando $\gamma^{(i)} = m_\theta(x^{(i)})\left(1-m_\theta(x^{(i)})\right)$, obtém-se

$$\frac{\partial^2 f}{\partial\theta_k\partial\theta_j}(\theta) = \sum_{i=1}^{s}x_j^{(i)}x_k^{(i)}\gamma^{(i)}.$$

Portanto,

$$\nabla^2 f(\theta) = X^T\Gamma X,$$

onde $\Gamma = \mathrm{diag}(\gamma^{(1)},\ldots,\gamma^{(s)})$, e X é dado como em (2.11). Como

$$\gamma^{(i)} = m_\theta(x^{(i)})\left(1-m_\theta(x^{(i)})\right) > 0,$$

para $i = 1,\ldots,s$, segue-se que

$$z^T\nabla^2 f(\theta)z = z^T X^T\Gamma Xz = (Xz)^T\Gamma Xz.$$

Então, fazendo $Xz = w$, obtém-se

$$w^T\Gamma w = \sum_{i=1}^{s}\gamma^{(i)}\left(w^{(i)}\right)^2 \geq 0.$$

Disto segue que $\nabla^2 f(\theta)$ é semidefinida positiva, e por conseguinte, f é uma função convexa. ■

Observação 2.6. *Se X tem posto completo, então $Xz = w \neq 0$ para todo z, e a Hessiana de f é definida positiva. Neste caso, f é estritamente convexa.*

Outra propriedade importante no contexto da otimização é o fato de que ∇f é Lipschitz.

Teorema 2.7. *O gradiente da função dada em (2.9) é L-Lipschitz, com $L = \|X\|^2$.*

Demonstração: Pela Desigualdade do Valor Médio, existe $\bar{\theta} \in (\theta_1, \theta_2)$ [2] tal que

$$\|\nabla f(\theta_1) - \nabla f(\theta_2)\| \leq \left\|\nabla^2 f(\bar{\theta})\right\| \|\theta_1 - \theta_2\|.$$

Como

$$\nabla^2 f(\theta) = X^T \Gamma X,$$

e Γ é uma matriz diagonal cujos elementos tem módulo menor que 1, tem-se que

$$\|\nabla f(\theta_1) - \nabla f(\theta_2)\| \leq \left\|X^T\right\| \|\Gamma\| \|X\| \|\theta_1 - \theta_2\| \leq \|X\|^2 \|\theta_1 - \theta_2\|.$$

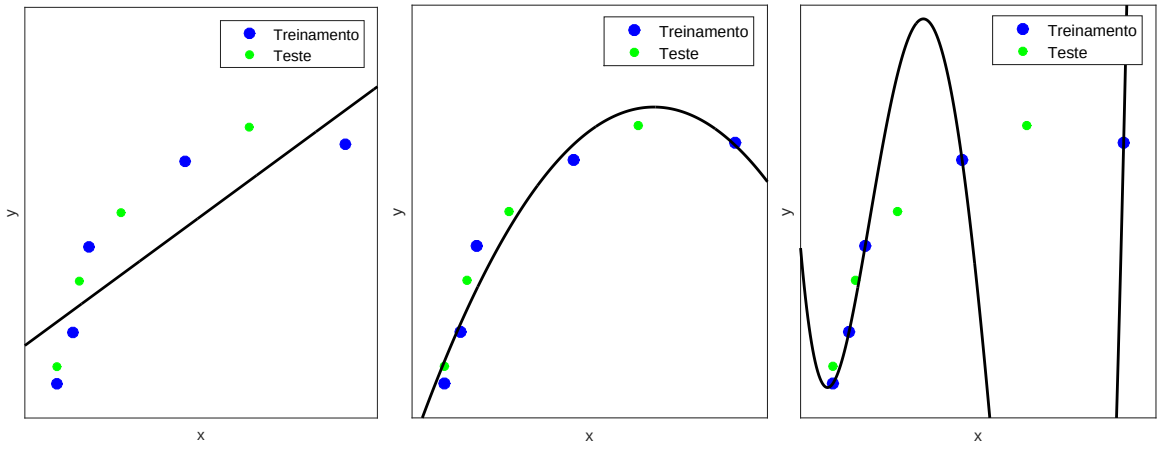
Logo, $\nabla f(\theta)$ é L-Lipschitz, com $L = \|X\|^2$. ∎

Sendo uma função convexa com gradiente Lipschitz, a função objetivo f em (2.9) pode ser minimizada usando-se, por exemplo, os métodos de descida e o método do gradiente acelerado descritos no Capítulo 1. Usando como critério a complexidade de pior caso, a preferência de escolha seria para o método acelerado.

Assim como o problema de Regressão Linear, o problema de Regressão Logística também admite adaptações. Uma modificação interessante é a introdução de um termo de regularização $\frac{\lambda}{2}\|\theta\|^2$, com $\lambda > 0$. Isto resulta no problema regularizado

$$\min_{\theta \in \mathbb{R}^{n+1}} f(\theta) = -\sum_{i=1}^{s} \left[y^{(i)} \log\left(m_\theta(x^{(i)})\right) + (1 - y^{(i)}) \log\left(1 - m_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2} \sum_{j=0}^{n} \theta_j^2. \quad (2.12)$$

Tal regularização tem por objetivo evitar erros de underfitting e overfitting [3]. O underfit-



(a) Underfitting (b) Ajuste ideal (c) Overfitting

Figura 2.5: Exemplo de erros de underfitting e overfitting.

[2] Dados $\theta_1, \theta_2 \in \mathbb{R}^n$, define-se $(\theta_1, \theta_2) = \{\theta_1 + t(\theta_2 - \theta_1); 0 < t < 1\}$

[3] A utilização de outras normas também é válida, e pode ser mais eficiente. A norma l_1, por exemplo, pode ser mais eficaz para eliminar variáveis menos importantes.

ting é caracterizado pela incapacidade do modelo de se ajustar aos dados de treinamento (conjunto de dados utilizado para treinar o modelo) e, consequentemente, aos dados de teste (conjunto de dados utilizado para avaliar a performance do modelo treinado na realização de previsões); já no overfitting o modelo tende a se ajustar demais aos dados de treinamento, perdendo sua capacidade de generalização. A Figura 2.5 ilustra um exemplo com underfitting, ajuste ideal e overfitting, respectivamente.

2.3 Máquinas de Vetor Suporte

As Máquinas de Vetor Suporte (SVM, do inglês Support Vector Machines) também são utilizadas em problemas de classificação, assim como a Regressão Logística. Entretanto, SVM não tem caráter probabilístico. Além disso, enquanto na Regressão Logística procura-se simplesmente por um hiperplano que separe os dados, no SVM busca-se um hiperplano que separe os dados com a maior folga possível, ou seja, de modo que a distância entre o hiperplano separador e os pontos mais próximos a ele seja a maior possível. Esta ideia foi introduzida por Vapnik e Lerner em 1963 [28], e complementada por Vapnik e outros colaboradores nos anos seguintes [27, 5, 7]. Ela está ilustrada na Figura 2.6. Observe que todas as retas separam o conjunto de dados, mas a linha contínua é a que mantém a maior distância dos pontos. Tal linha seria a reta do modelo SVM, também chamada de hiperplano ótimo.

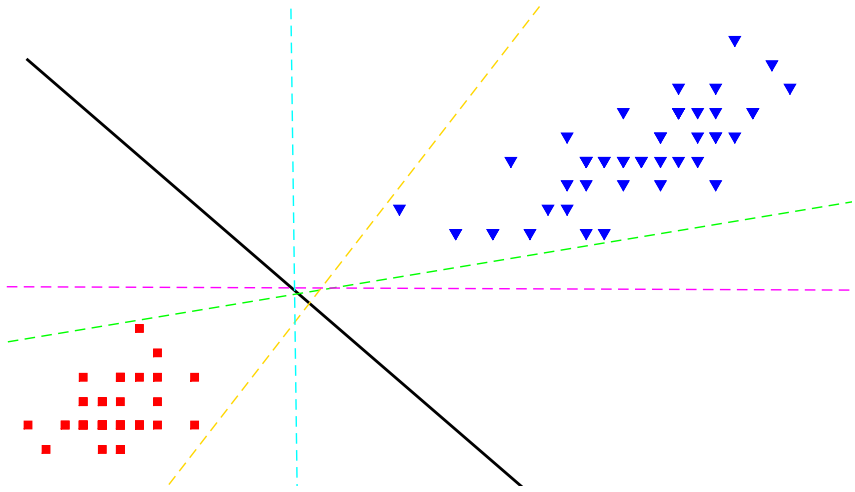


Figura 2.6: Hiperplano separador SVM.

Considere um conjunto de dados

$$\Omega = \left\{ \left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(s)}, y^{(s)}\right) \right\},$$

onde $x^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)}) \in \mathbb{R}^n$ e $y^{(i)} \in \{-1, 1\}$, para $i = 1, \ldots, s$. O problema de treinamento do modelo SVM pode ser deduzido a partir do problema de treinamento do

modelo de regressão logística. Para regressão logística com regularização, o parâmetro Θ do hiperplano separador $\Theta^T x = 0$ é obtido resolvendo-se o problema (2.12), com $m_\Theta(x) = \frac{1}{1+e^{-\Theta^T x}}$, $\lambda > 0$ e $y^{(i)} = -1$ substituído por $y^{(i)} = 0$. O problema (2.12) pode ser modificado para

$$\min_{\theta \in \mathbb{R}^n, b \in \mathbb{R}} f(\theta, b) = C \sum_{i=1}^{s} \left[y^{(i)} \tilde{c}_1(\theta^T x^{(i)} + b) + (1 - y^{(i)}) \tilde{c}_0(\theta^T x^{(i)} + b) \right] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2, \quad (2.13)$$

com $C = \frac{1}{\lambda}$,

$$\tilde{c}_1(z) = -\log\left(\frac{1}{1+e^{-z}}\right) \quad \text{e} \quad \tilde{c}_0(z) = -\log\left(1 - \frac{1}{1+e^{-z}}\right).$$

Sob esse ponto de vista, o problema de treinamento do modelo SVM pode ser obtido substituindo-se as funções $\tilde{c}_1$ e $\tilde{c}_0$ em (2.13) por funções convenientes. Especificamente, considere funções

$$c_1(z) = \begin{cases} -a_1 z + b_1, & \text{se } z < 1 \\ 0, & \text{se } z \geq 1 \end{cases}$$

e

$$c_0(z) = \begin{cases} a_0 z + b_0, & \text{se } z > -1 \\ 0, & \text{se } z \leq -1 \end{cases},$$

com $a_1, a_0 > 0$ e $b_1, b_0 \in \mathbb{R}$ tais que os gráficos de c_1 e c_0 se comportem conforme ilustrado na Figura 2.7.

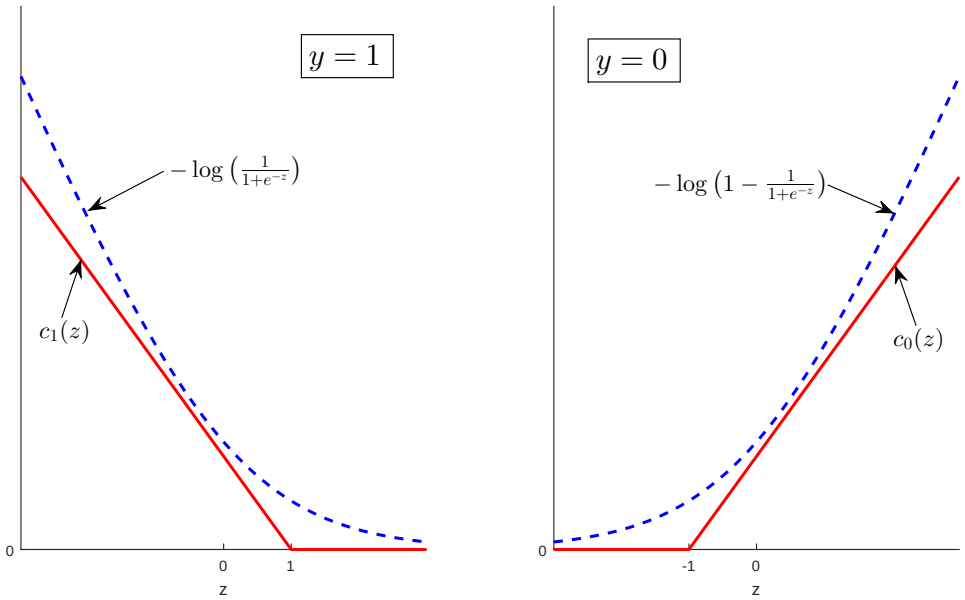


Figura 2.7: Funções para Regressão Logística e para SVM.

Tem-se então o seguinte problema de otimização:

$$\min_{\theta\in\mathbb{R}^n, b\in\mathbb{R}} f(\theta, b) = C\sum_{i=1}^{s}\left[y^{(i)}c_1(\theta^T x^{(i)} + b) + (1 - y^{(i)})c_0(\theta^T x^{(i)} + b)\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2. \quad (2.14)$$

Neste caso, o ideal seria encontrar θ e b tal que

$$\sum_{i=1}^{s}\left[y^{(i)}c_1(\theta^T x^{(i)} + b) + (1 - y^{(i)})c_0(\theta^T x^{(i)} + b)\right] = 0.$$

Com base nas definições de c_1 e c_0, isto ocorre se, e somente se

$$\begin{cases} \theta^T x^{(i)} + b \geq 1, & \text{quando } y^{(i)} = 1 \\ \theta^T x^{(i)} + b \leq -1, & \text{quando } y^{(i)} = 0 \end{cases},$$

para $i = 1, \ldots, s$. Assim, impondo-se tal condição, o problema (2.14) se reduz a

$$\begin{aligned} &\min_{\theta\in\mathbb{R}^n} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 \\ &s.a.\ \theta^T x^{(i)} + b \geq 1, \qquad \text{se } y^{(i)} = 1 \\ &\qquad \theta^T x^{(i)} + b \leq -1, \quad \text{se } y^{(i)} = 0. \end{aligned} \quad (2.15)$$

Nesta formulação, os valores que $y^{(i)}$ assume não influenciam na resolução do problema, bastando que $y^{(i)}$ seja uma variável binária. Convém então retornar ao caso inicial $y^{(i)} \in \{-1, 1\}$, pois assim (2.15) pode ser reescrito na forma canônica do problema de treinamento do modelo SVM:

$$\begin{aligned} &\min_{\theta\in\mathbb{R}^n} \frac{1}{2}\|\theta\|^2 \\ &s.a.\ y^{(i)}(\theta^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, s. \end{aligned} \quad (2.16)$$

Uma vez obtido θ pela resolução de (2.16), tem-se então três hiperplanos: o hiperplano separador $H : \left\{x \mid \theta^T x + b = 0\right\}$, e dois outros hiperplanos delimitando a margem de ambos os lados, que são $H_1 : \left\{x \mid \theta^T x + b = 1\right\}$ e $H_2 : \left\{x \mid \theta^T x + b = -1\right\}$.

Lema 2.8. *A distância entre os hiperplanos H_1 e H_2 é igual a*

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

Demonstração: Considere $x_1 \in H_1$ e $x_2 \in H_2$. A distância entre H_1 e H_2 é dada pela norma da projeção ortogonal do vetor $(x_1 - x_2)$ em θ, ou seja,

$$d(H_1, H_2) = \|proj_\theta(x_1 - x_2)\|.$$

Como

$$proj_\theta(x_1 - x_2) = \frac{\theta^T(x_1 - x_2)}{\theta^T\theta}\theta,$$

tem-se que

$$\|proj_\theta(x_1 - x_2)\| = \frac{\left|\theta^T(x_1 - x_2)\right|}{\|\theta\|}.$$

Além disso,

$$\theta^T x_1 + b = 1 \quad \text{e} \quad \theta^T x_2 + b = -1,$$

de onde

$$\theta^T(x_1 - x_2) = 2.$$

Portanto,

$$d(H_1, H_2) = \|proj_\theta(x_1 - x_2)\| = \frac{2}{\|\theta\|}.$$

■

Corolário 2.9. *Resolver o problema de otimização* (2.16) *é equivalente a encontrar o hiperplano que separa os dados com maior margem possível.*

Demonstração: Encontrar o hiperplano com maior margem possível é o mesmo que obter H_1 e H_2 com máxima distância entre si, de forma que a separação do conjunto de dados Ω seja feita corretamente, ou seja, que as amostras com $y^{(i)} = 1$ fiquem de um lado, e as amostras com $y^{(i)} = -1$ fiquem do outro, supondo que isto seja possível. Ora, aumentar a distância entre H_1 e H_2 é equivalente a minimizar $\|\theta\|^2$, uma vez que, pelo Teorema 2.8, $d(H_1, H_2) = \frac{2}{\|\theta\|}$. Para que as amostras fiquem devidamente classificadas, impõe-se a condição de que

$$y^{(i)}\left(\theta^T x^{(i)} + b\right) \geq 1, \quad i = 1, \ldots, s.$$

Portanto, o problema de encontrar o hiperplano com maior margem possível pode ser resolvido a partir do seguinte problema de otimização:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^n} \quad & \frac{1}{2}\|\theta\|^2 \\ s.a. \quad & y^{(i)}(\theta^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, s \end{aligned}$$

■

Definição 2.10. *Os pontos que satisfazem* $y^{(i)}(\theta^T x^{(i)} + b) = 1$ *são denominados* ***vetores suporte****.*

Definição 2.11. *Um conjunto de dados é dito linearmente separável quando pode ser separado por um hiperplano.*

A Figura 2.8 ilustra um conjunto linearmente separável e outro não linearmente separável.

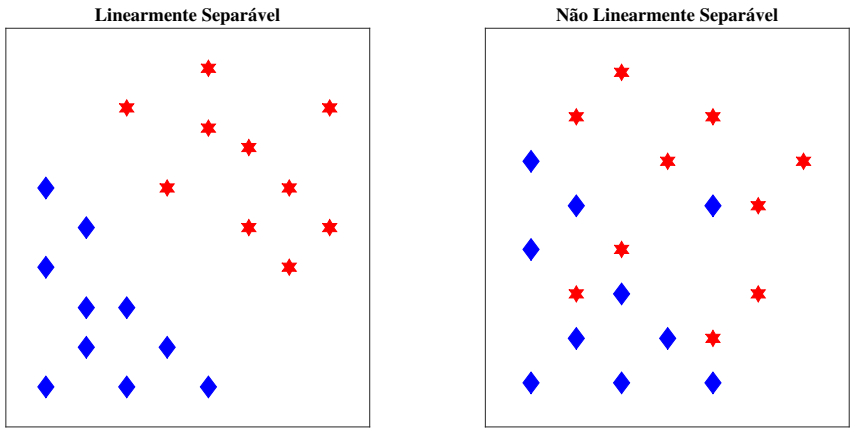


Figura 2.8: Exemplos de conjuntos linearmente e não linearmente separáveis.

Os problemas práticos em geral não são linearmente separáveis. Para tentar contornar este fato, pode-se utilizar a versão de SVM com margens flexíveis, ou C-SVM, que permite que alguns pontos infrinjam a margem, acrescentando-se algumas variáveis de folga. Neste caso, o problema de otimização a ser resolvido é

$$\begin{aligned} \min_{\theta\in\mathbb{R}^n,\xi\in\mathbb{R}^s} \quad & \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{s}\xi_i \\ s.a. \quad & y^{(i)}\left(\theta^T x^{(i)} + b\right) \geq 1-\xi_i, \quad i=1,\ldots,s \\ & \xi_i \geq 0, \quad i=1,\ldots,s, \end{aligned}$$

onde $C > 0$ é um parâmetro determinado experimentalmente. Quanto maior o valor de C, menor é a tolerância do método a violações da margem, ou seja, mais rígida, e consequentemente menor ela tende a ser.

Existem ainda situações em que a separação linear não é a melhor opção para o conjunto de dados. Nestes casos, a estratégia consiste em levar os dados a um espaço de maior dimensão, onde estes sejam linearmente separáveis. Isto é feito com a utilização de núcleos (kernels), os quais não serão abordados neste trabalho.

2.4 Redes Neurais Artificiais

Redes Neurais Artificiais são modelos matemáticos inspirados pelo cérebro humano. Sabe-se que o cérebro é uma estrutura extremamente complexa, sendo composto por uma quantidade gigantesca de células, denominadas neurônios. Por sua vez, cada neurônio pode ser dividido em três partes: os dendritos, que recebem estímulos; o corpo celular, responsável pelo processamento das informações recebidas; e o axônio, pelo qual impulsos são trans-

mitidos para outros neurônios. A Figura 2.9 [4] contém uma representação simplificada de um neurônio biológico.

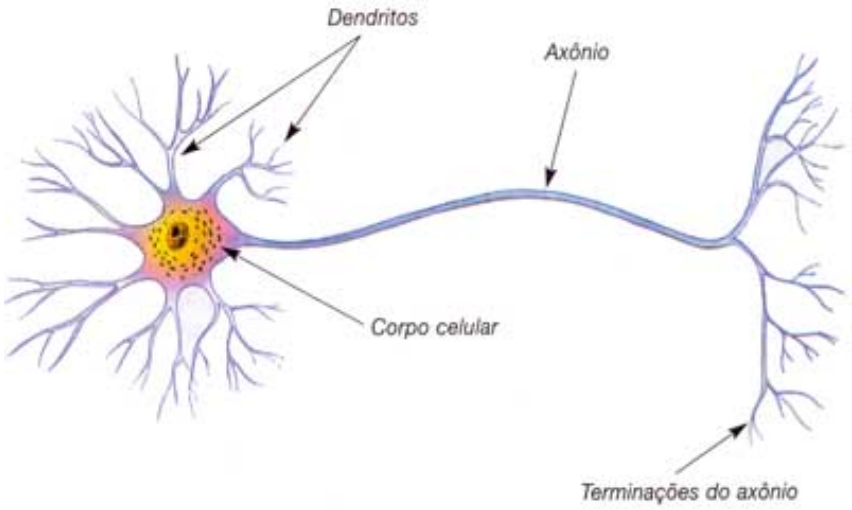


Figura 2.9: Modelo de neurônio biológico.

No cérebro, os neurônios se conectam uns aos outros por meio de sinapses, formando assim uma grande rede capaz de realizar tarefas altamente complexas. Tomando essa estrutura como referência, uma Rede Neural Artificial consiste num conjunto de elementos simples, também denominados neurônios, capazes de processar e transmitir informações, e que conectados podem exercer funções mais complexas.

O processo básico é o seguinte: um neurônio recebe informações de outros neurônios através das conexões (sinapses) existentes entre eles. A cada conexão entre os neurônios está associado um peso, denominado peso sináptico. Estas informações são sintetizadas e processadas dentro do neurônio, passando por uma função de ativação, que determina uma saída a ser emitida pelo mesmo. Esta saída pode servir como entrada para outro neurônio, e o processo se repete, até se obter a saída final. A estrutura básica de um neurônio artificial é ilustrada na Figura 2.10.

Há muitas possibilidades para a estrutura de Redes Neurais Artificiais. Aqui, o foco é no modelo da forma *Feedforward* de múltiplas camadas (Multilayer). O termo *Feedforward* significa que um sinal se propaga apenas no sentido entrada-saída. As camadas consideradas são: camada de Entrada (Input layer), camadas Intermediárias ou Escondidas (Hidden layers) e camada de Saída (Output layer). Cada camada pode possuir diferentes quantidades de neurônios. Este tipo de rede neural está representado na Figura 2.11.

A camada de entrada apenas recebe os dados de entrada e os transmite para a primeira camada intermediária, onde os dados são processados por neurônios, produzindo sinais

[4]Fonte: `http://www.sobiologia.com.br/conteudos/FisiologiaAnimal/nervoso2.php` (Acessado em 10 de Agosto de 2017).

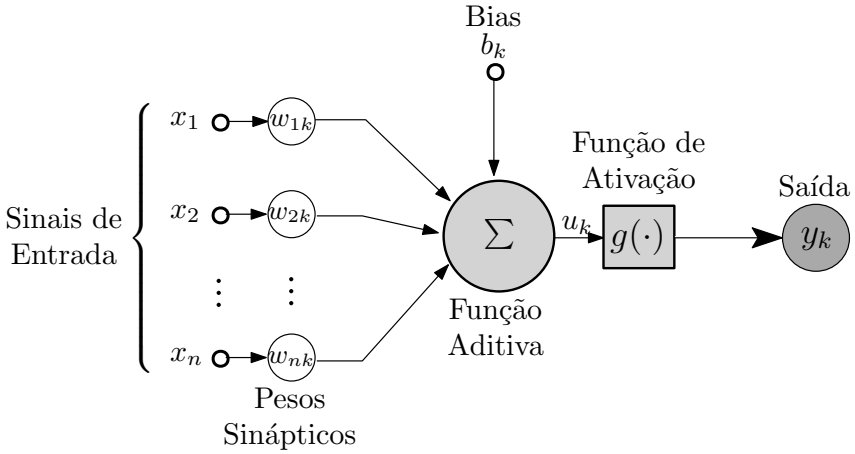


Figura 2.10: Modelo de neurônio artificial.

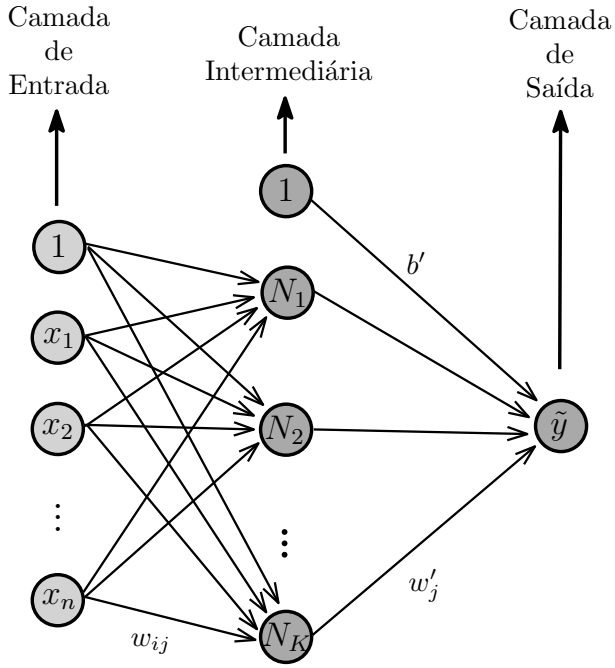


Figura 2.11: Representação de Rede Neural Feedforward Multicamadas.
Note que neste caso o bias está representado pelo neurônio com valor 1.

que são transmitidos para a camada seguinte, e o processo é repetido até chegar à camada de saída.

Após determinar a estrutura da rede – quantidade de camadas e quantidade de neurônios em cada camada, o problema torna-se decidir quais são os pesos sinápticos apropriados para que a rede neural processe os dados de entrada de modo satisfatório. Para isto, resolve-se um problema de otimização, buscando minimizar o erro entre a saída gerada pela rede e a saída desejada. A função erro utilizada pode ser a soma dos erros quadráticos (SEQ) ou a entropia cruzada (EC).

Considere um conjunto de dados

$$\Omega = \left\{ \left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(s)}, y^{(s)}\right) \right\},$$

onde $x^{(i)} = (x_0^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)}) \in \mathbb{R}^{n+1}$, com $x_0^{(i)} = 1$ e $y^{(i)} \in \mathbb{R}$, $i = 1, \ldots, s$.

A função erro SEQ é dada por

$$f(\theta) = \frac{1}{2} \sum_{i=1}^{s} \left(y^{(i)} - \tilde{y}^{(i)}\right)^2 \tag{2.17}$$

onde θ representa o vetor cujas entradas são os pesos sinápticos da rede, e $\tilde{y}^{(i)}$ é a saída gerada pela rede neural artificial para a entrada $x^{(i)}$ (observe que $\tilde{y}^{(i)}$ depende de θ).

Já a função erro EC é dada por

$$f(\theta) = -\sum_{i=1}^{s} \left[y^{(i)} \log(\tilde{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \tilde{y}^{(i)}) \right]. \tag{2.18}$$

Considere uma rede com uma única camada escondida com K neurônios, uma única saída (ou seja, apenas um neurônio na camada de saída), e funções de ativação g_1 na camada intermediária, e g_2 na camada de saída. Cada neurônio N_j da camada escondida recebe uma combinação linear das componentes da entrada[5] e aplica a função de ativação g_1, devolvendo uma saída da forma

$$N_j = g_1 \left(\sum_{i=0}^{n} w_{ij} x_i \right). \tag{2.19}$$

A camada de saída recebe, então, uma combinação linear de tais N_j somada a um bias b', e aplica a função de ativação g_2, gerando a saída

$$\tilde{y} = g_2 \left(\sum_{j=1}^{K} w'_j N_j + b' \right).$$

Segundo Bishop [2](pág. 236), há uma escolha natural para a função erro e a função de ativação na camada de saída (g_2), de acordo com o tipo de problema a ser resolvido. Para problemas de regressão, utiliza-se SEQ como função erro e uma função de ativação linear na camada de saída, dada por $g_2(z) = z$. Já para classificação binária, considera-se a função erro EC e função de ativação logística, ou seja, $g_2(z)$ é dada pela equação (2.7).

Enfim, determinada a estrutura da rede e a função erro f, resolve-se o problema de

[5] Aqui, cada entrada é da forma $x = (x_0, x_1, \ldots, x_n)$, onde $x_0 = 1$ faz o papel de bias. Outra forma de expressar (2.19) seria $N_j = g_1 \left(\sum_{i=1}^{n} w_{ij} x_i + b_j \right)$.

otimização

$$\min_{\theta} \ f(\theta)$$

para determinar os pesos sinápticos da rede. Tais pesos são dados por w_{ij}, w'_j e b', com $i = 0, \ldots, n$ e $j = 1, \ldots, K$, e ficam armazenados no vetor θ. Como a maioria dos métodos de otimização envolvem o gradiente da função, seria interessante obtê-lo. Tem-se que f é dada por (2.17) ou (2.18). O gradiente destas funções pode ser calculado através do método *Backpropagation*, que consiste em sucessivas aplicações da Regra da Cadeia.

Note que cada função erro é um somatório de funções, que pode ser reescrito como

$$f(\theta) = \sum_{i=1}^{s} f^{(i)}(\theta),$$

onde

$$f^{(i)}(\theta) = \begin{cases} \frac{1}{2}\left(y^{(i)} - \tilde{y}^{(i)}\right)^2, & \text{para SEQ} \\ -\left[y^{(i)} \log(\tilde{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \tilde{y}^{(i)})\right], & \text{para EC.} \end{cases}$$

Consequentemente,

$$\nabla f(\theta) = \sum_{i=1}^{s} \nabla f^{(i)}(\theta).$$

Nos teoremas a seguir considera-se $s = 1$, sem perda de generalidade.

Teorema 2.12. *Considere uma rede neural como mencionada anteriormente, com $g_2(z) = z$. Se a função erro é dada por* (2.17)*, tem-se as seguintes derivadas parciais:*

$$\frac{\partial f}{\partial w_{ij}}(\theta) = (\tilde{y} - y) w'_j g'_1(v_j) x_i$$

$$\frac{\partial f}{\partial w'_j}(\theta) = (\tilde{y} - y) N_j$$

$$\frac{\partial f}{\partial b'}(\theta) = (\tilde{y} - y)$$

onde

$$v_j = \sum_{i=0}^{n} w_{ij} x_i.$$

Demonstração: Como $s = 1$, tem-se que

$$f(\theta) = \frac{1}{2}(\tilde{y} - y)^2.$$

Para obter $\nabla f(\theta)$, deve-se calcular as derivadas parciais

$$\frac{\partial f}{\partial w_{ij}}(\theta), \ \frac{\partial f}{\partial w'_j}(\theta), \ \text{e} \ \frac{\partial f}{\partial b'}(\theta).$$

Inicialmente, observe que

$$\nabla f(\theta) = (\tilde{y} - y)\nabla\tilde{y}(\theta), \tag{2.20}$$

uma vez que apenas $\tilde{y}$ é dependente de θ. Então,

1. $$\frac{\partial f}{\partial w_{ij}}(\theta) = (\tilde{y} - y)\frac{\partial \tilde{y}}{\partial w_{ij}}(\theta).$$

   Como $\tilde{y} = g_2\left(\sum_{k=1}^{K} w'_k N_k + b'\right)$, tem-se

   $$\frac{\partial \tilde{y}}{\partial w_{ij}}(\theta) = \frac{\partial g_2}{\partial w_{ij}}(\theta) = \frac{\partial g_2}{\partial u}\frac{\partial u}{\partial w_{ij}}(\theta),$$

   onde $u = \sum_{k=1}^{K} w'_k N_k + b'$, e portanto

   $$\frac{\partial u}{\partial w_{ij}}(\theta) = \sum_{k=1}^{K} w'_k \frac{\partial N_k}{\partial w_{ij}}(\theta).$$

   Como $N_k = g_1\left(\sum_{i=0}^{n} w_{ik}x_i\right)$, tem-se ainda

   $$\frac{\partial N_k}{\partial w_{ij}}(\theta) = \frac{\partial g_1}{\partial v_k}\frac{\partial v_k}{\partial w_{ij}}(\theta),$$

   onde $v_k = \sum_{i=0}^{n} w_{ik}x_i$, e portanto

   $$\frac{\partial v_k}{\partial w_{ij}}(\theta) = \begin{cases} 0, & \text{se } k \neq j \\ x_i, & \text{se } k = j \end{cases}$$

   Assim,

   $$\frac{\partial f}{\partial w_{ij}}(\theta) = (\tilde{y} - y)g'_2(u)w'_j g'_1(v_j)x_i.$$

2. $$\frac{\partial f}{\partial w'_j}(\theta) = (\tilde{y} - y)\frac{\partial \tilde{y}}{\partial w'_j}(\theta).$$

   Como $\tilde{y} = g_2(u) = g_2\left(\sum_{j=1}^{K} w'_j N_j + b'\right)$, tem-se

   $$\frac{\partial \tilde{y}}{\partial w'_j}(\theta) = \frac{\partial g_2}{\partial u}\frac{\partial u}{\partial w'_j}(\theta) = g'_2(u)N_j,$$

   e portanto

   $$\frac{\partial f}{\partial w'_j}(\theta) = (\tilde{y} - y)g'_2(u)N_j.$$

3. $$\frac{\partial f}{\partial b'}(\theta) = (\tilde{y} - y)\frac{\partial \tilde{y}}{\partial b'}(\theta) = (\tilde{y} - y)\frac{\partial g_2}{\partial u}\frac{\partial u}{\partial b'}(\theta) = (\tilde{y} - y)g'_2(u).$$

Se $g_2(z) = z$, segue que $g_2'(z) = 1$. Fazendo esta substituição nas derivadas parciais obtidas, conclui-se a demonstração. ∎

Lema 2.13. *Se $g(z)$ é a função logística, então*

$$g'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = g(z)\,(1-g(z))\,.$$

Demonstração: De fato,

$$g(z)\,(1-g(z)) = \frac{1}{1+e^{-z}}\left(1-\frac{1}{1+e^{-z}}\right) = \frac{1}{1+e^{-z}}\left(\frac{e^{-z}}{1+e^{-z}}\right) = \frac{e^{-z}}{(1+e^{-z})^2}.$$

∎

Teorema 2.14. *Considere uma rede neural como mencionada anteriormente, com g_2 sendo a função logística. Se a função erro é dada por* (2.18)*, tem-se as seguintes derivadas parciais:*

$$\frac{\partial f}{\partial w_{ij}}(\theta) = (\tilde{y}-y)\,w_j' g_1'(v_j)x_i$$

$$\frac{\partial f}{\partial w_j'}(\theta) = (\tilde{y}-y)\,N_j$$

$$\frac{\partial f}{\partial b'}(\theta) = (\tilde{y}-y)$$

onde

$$v_j = \sum_{i=0}^{n} w_{ij}x_i.$$

Demonstração: Como $s = 1$, tem-se

$$f(\theta) = -\left[y\log(\tilde{y}) + (1-y)\log(1-\tilde{y})\right].$$

De modo análogo ao que obteve-se em (2.20), para esta função tem-se que

$$\begin{aligned}
\nabla f(\theta) &= \frac{\partial f}{\partial \tilde{y}}\nabla\tilde{y}(\theta) \\
&= -\left[y\frac{1}{\tilde{y}}\nabla\tilde{y}(\theta) + (1-y)\frac{1}{1-\tilde{y}}\left(-\nabla\tilde{y}(\theta)\right)\right] \\
&= -\frac{y(1-\tilde{y}) - (1-y)\tilde{y}}{\tilde{y}(1-\tilde{y})}\nabla\tilde{y}(\theta) \\
&= \frac{\tilde{y}-y}{\tilde{y}(1-\tilde{y})}\nabla\tilde{y}(\theta).
\end{aligned}$$

Como $\tilde{y}$ é calculado da mesma maneira nos dois modelos, apenas variando as funções de ativação g_1 e g_2, segue que $\nabla\tilde{y}(\theta)$ pode ser obtido de modo análogo à demonstração do teorema anterior. Assim, obtém-se as seguintes derivadas parciais:

$$\frac{\partial f}{\partial w_{ij}}(\theta) = \frac{\tilde{y} - y}{\tilde{y}(1-\tilde{y})} g_2'(u) w_j' g_1'(v_j) x_i$$

$$\frac{\partial f}{\partial w_j'}(\theta) = \frac{\tilde{y} - y}{\tilde{y}(1-\tilde{y})} g_2'(u) N_j$$

$$\frac{\partial f}{\partial b'}(\theta) = \frac{\tilde{y} - y}{\tilde{y}(1-\tilde{y})} g_2'(u).$$

Neste caso, entretanto, $g_2(z)$ é a função logística, e portanto, pelo Lema 2.13, $g_2'(z) = g_2(z)\,(1 - g_2(z))$. Pelo primeiro item da demonstração anterior, tem-se que $g_2(u) = \tilde{y}$. Assim, pode-se substituir $g_2'(u)$ por $\tilde{y}(1-\tilde{y})$, obtendo as seguintes derivadas parciais para esta situação:

$$\frac{\partial f}{\partial w_{ij}}(\theta) = \frac{\tilde{y} - y}{\tilde{y}(1-\tilde{y})} \tilde{y}(1-\tilde{y}) w_j' g_1'(v_j) x_i = (\tilde{y} - y) w_j' g_1'(v_j) x_i$$

$$\frac{\partial f}{\partial w_j'}(\theta) = \frac{\tilde{y} - y}{\tilde{y}(1-\tilde{y})} \tilde{y}(1-\tilde{y}) N_j = (\tilde{y} - y) N_j$$

$$\frac{\partial f}{\partial b'}(\theta) = \frac{\tilde{y} - y}{\tilde{y}(1-\tilde{y})} \tilde{y}(1-\tilde{y}) = (\tilde{y} - y).$$

■

Neste trabalho, considera-se $g_1(z)$ como sendo a função logística. Neste caso, as derivadas parciais dos Teoremas 2.12 e 2.14 (note que são iguais) são dadas por

$$\frac{\partial f}{\partial w_{ij}}(\theta) = (\tilde{y} - y) w_j' N_j (1 - N_j) x_i$$

$$\frac{\partial f}{\partial w_j'}(\theta) = (\tilde{y} - y) N_j$$

$$\frac{\partial f}{\partial b'}(\theta) = (\tilde{y} - y).$$

Tendo isto, pode-se utilizar qualquer método envolvendo o gradiente para encontrar θ que minimize a função erro e, tendo os pesos sinápticos da rede, prever resultados para amostras cuja saída seja desconhecida. É importante ressaltar, entretanto, que em geral a função a ser minimizada não é convexa. Portanto, não se tem a garantia de obter um minimizador global do problema.

O Teorema da Aproximação Universal estabelece que a rede neural descrita anteriormente, ou seja, com uma única camada escondida e única saída, pode aproximar qualquer função contínua definida sobre um conjunto compacto de $\mathbb{R}^n$.

Definição 2.15. *Uma função g é sigmoidal se* $g(t) \to \begin{cases} 1, & \text{quando } t \to \infty \\ 0, & \text{quando } t \to -\infty \end{cases}$.

Note que a função logística é um exemplo de função sigmoidal.

Teorema 2.16 (Teorema da Aproximação Universal)**.** *Seja g uma função sigmoidal contínua, e $\Omega \subset \mathbb{R}^n$ um conjunto compacto. Dada $f \in \mathcal{C}(\Omega)$, e dado qualquer $\varepsilon > 0$, existem $N \in \mathbb{N}$, w_i', $b_i \in \mathbb{R}$ e $w_i \in \mathbb{R}^n$, para $i = 1, 2, ..., N$, tais que a função $F : \mathbb{R}^n \to \mathbb{R}$ dada por*

$$F(x) = \sum_{i=1}^{N} w_i' g(w_i^T x + b_i)$$

satisfaz $|F(x) - f(x)| < \varepsilon$.

Demonstração: Ver Cybenko [8]. ∎

Capítulo 3

Aprendizagem de Máquina e a previsão dos movimentos do Ibovespa

O Ibovespa resulta de uma seleção de ativos que satisfazem certos critérios estabelecidos pela BM&FBovespa. O objetivo do Ibovespa é "ser o indicador do desempenho médio das cotações dos ativos de maior negociabilidade e representatividade do mercado de ações brasileiro"[3]. A carteira teórica do Ibovespa é atualizada a cada 4 meses. No primeiro quadrimestre de 2017, por exemplo, as empresas com maior participação no Ibovespa eram Itau Unibanco, Bradesco, Ambev, Vale e Petrobras, num total de aproximadamente 60 ações.

Este capítulo descreve a aplicação dos modelos de Aprendizagem de Máquina abordados no Capítulo 2 ao problema de prever os movimentos de alta ou baixa do Ibovespa. O conjunto de dados considerado consiste na série histórica mensal do Ibovespa cobrindo um período de 15 anos: de janeiro de 2002 a dezembro de 2016. Esta série, ilustrada na Figura 3.1, foi obtida no site do Yahoo Finanças [30].

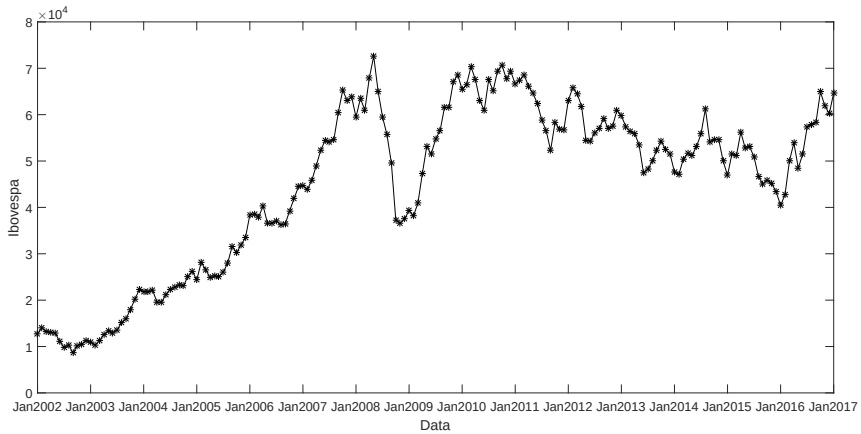


Figura 3.1: Série Ibovespa.

3.1 Previsão vista como problema de Classificação

Considere a série temporal $\{d_i\}_{i=1}^{T}$, onde d_i é o valor do Ibovespa no mês i. Fixado $h \in \{1, 3, 6, 12\}$, para todo $i \in [6, T-h] \cap \mathbb{N}$ deseja-se obter uma previsão para

$$y^{(i)} = \text{sinal}(d_{i+h} - d_i),$$

tendo como variáveis explicativas os valores

$$d_i,\ d_{i-1},\ d_{i-2},\ d_{i-3},\ d_{i-4},\ d_{i-5}.$$

Ou seja, o objetivo no mês i é prever se o valor do Ibovespa após h meses será maior ou menor que seu valor corrente, usando-se para isso os valores do índice nos últimos 6 meses.

Para cada $i \in [6, T-h] \cap \mathbb{N}$, defina

$$x^{(i)} = (d_{i-5},\ d_{i-4},\ d_{i-3},\ d_{i-2},\ d_{i-1},\ d_i) \in \mathbb{R}^6,$$

e considere o conjunto $\Omega = \left\{\left(x^{(6)}, y^{(6)}\right), \ldots, \left(x^{(T-h)}, y^{(T-h)}\right)\right\} \subset \mathbb{R}^6 \times \{-1, 1\}$. Dessa forma, pode-se reduzir o problema de previsão dos movimentos do Ibovespa ao problema de classificação binária dos pontos de Ω.

Para as implementações computacionais, fixado h, os dados foram sintetizados em uma matriz de entradas

$$X = \begin{bmatrix} 1 & x_1^{(6)} & \ldots & x_6^{(6)} \\ 1 & x_1^{(7)} & \ldots & x_6^{(7)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(T-h)} & \ldots & x_6^{(T-h)} \end{bmatrix} \in \mathbb{R}^{(T-h-5)\times 7}$$

e um vetor de saídas

$$y = \begin{bmatrix} y^{(6)} \\ y^{(7)} \\ \vdots \\ y^{(T-h)} \end{bmatrix} \in \mathbb{R}^{(T-h-5)}.$$

O treinamento e teste dos modelos se deu utilizando-se, respectivamente, 80% e 20% dos dados. Essa divisão dos dados foi feita particionando-se X e y da seguinte maneira:

$$X = \begin{bmatrix} X_{treino} \\ X_{teste} \end{bmatrix} \quad \text{e} \quad y = \begin{bmatrix} y_{treino} \\ y_{teste} \end{bmatrix}.$$

A quantidade de linhas da matriz X_{treino} é denotada por $|treino|$, e corresponde à quanti-

dade de elementos no conjunto de treinamento. Analogamente, a quantidade de linhas da matriz X_{teste} é denotada por $|teste|$, e corresponde à quantidade de elementos no conjunto de teste.

Destaca-se na Tabela 3.1 a quantidade de casos positivos e negativos no conjunto de teste, para cada valor de h considerado.

Tabela 3.1: Informações do conjunto de teste

h	Casos positivos	Casos negativos	Total
1	19	15	34
3	20	14	34
6	19	14	33
12	16	16	32

Para avaliar a eficácia de um dado modelo $\tilde{m}(x)$, após o treinamento do mesmo calcula-se $\tilde{y}^{(i)} = \tilde{m}(x^{(i)})$, para cada $x^{(i)}$ no conjunto de teste. O resultado recai em um dos três casos abaixo:

- Acerto: $\tilde{y}^{(i)} = y^{(i)}$;
- Falso-Positivo: $\tilde{y}^{(i)} = 1$ mas $y^{(i)} = -1$;
- Falso-Negativo: $\tilde{y}^{(i)} = -1$ mas $y^{(i)} = 1$.

A taxa de acerto (TA) do modelo $\tilde{m}$ é definida como

$$TA = \frac{\text{número de acertos}}{|teste|}.$$

Por fim, vale ressaltar que os resultados reportados neste capítulo e no Capítulo 4 foram obtidos usando o programa Matlab versão R2015a, com um computador utilizando processador Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz, memória RAM de 8GB, e sistema operacional Windows 10.

3.2 Regressão Logística

Conforme descrito na Seção 2.2, fixado o horizonte de previsão h, a determinação do parâmetro θ no modelo de Regressão Logística $m_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ é feita resolvendo problemas da forma

$$\min_{\theta \in \mathbb{R}^7} f(\theta) = -\sum_{i=1}^{s} \left[\bar{y}^{(i)} \log\left(m_\theta(x^{(i)})\right) + (1 - \bar{y}^{(i)}) \log\left(1 - m_\theta(x^{(i)})\right) \right] + \frac{\lambda}{2} \|\theta\|^2, \qquad (3.1)$$

onde $x^{(i)}$ é a i-ésima linha da matriz X_{treino}, $s = |treino|$, $\lambda \geq 0$ e $\bar{y}^{(i)}$ é dado por

$$\bar{y}^{(i)} = \begin{cases} 1, & \text{se } (y_{treino})_i = 1 \\ 0, & \text{se } (y_{treino})_i = -1. \end{cases} \tag{3.2}$$

Para cada $h \in \{1, 3, 6, 12\}$, o problema (3.1) correspondente foi resolvido considerando-se $\lambda = 0$, 10 e 100. Foram testados o Método de Newton Puro e o Método do Gradiente Acelerado descritos no Capítulo 1. Para ambos os métodos, a origem foi tomada como ponto inicial e a condição $\|\nabla f(\theta)\| < 10^{-6}$ foi usada como critério de parada. O Método de Newton mostrou-se consideravelmente mais rápido – em média 200 mil vezes.

Obtido o parâmetro θ, para cada $j = 1, \ldots, |teste|$ calculou-se

$$\tilde{y}^{(j)} = \begin{cases} 1, & \text{se } \theta^T x^{(j)} \geq 0 \\ -1, & \text{se } \theta^T x^{(j)} < 0 \end{cases},$$

com $x^{(j)}$ sendo a j-ésima linha da matriz X_{teste}. A Tabela 3.2 contém os resultados da comparação entre $\tilde{y}$ e y_{teste}. Em particular, a Figura 3.2 apresenta um comparativo com respeito às taxas de acerto.

Tabela 3.2: Resultados - Regressão Logística

Regularização	h	**Taxa de Acerto**	**Falso Positivo**	**Falso Negativo**
$\lambda = 0$	1	0.4118	0.3529	0.2353
	3	0.6176	0.2941	0.0882
	6	**0.6970**	0.2424	0.0606
	12	0.5313	0.4688	0.0000
$\lambda = 10$	1	0.4706	0.3235	0.2059
	3	0.6471	0.2647	0.0882
	6	0.6364	0.3030	0.0606
	12	0.4688	0.4063	0.1250
$\lambda = 100$	1	0.4706	0.3235	0.2059
	3	0.6471	0.2647	0.0882
	6	0.5758	0.3333	0.0909
	12	0.4375	0.3125	0.2500

A pior previsão obtida é para o horizonte de curtíssimo prazo $h = 1$ mês, com taxa de acerto inferior a 50%. Por outro lado, a melhor previsão ocorre para $h = 6$ meses com $\lambda = 0$, resultando em uma taxa de quase 70% de acerto.

3.3 C-SVM

Levando-se em conta a possibilidade dos dados não serem linearmente separáveis, optou-se pela utilização do modelo SVM com margens flexíveis (ou C-SVM). Conforme descrito na Seção 2.3, a determinação do parâmetro θ no modelo C-SVM é feita resolvendo-se

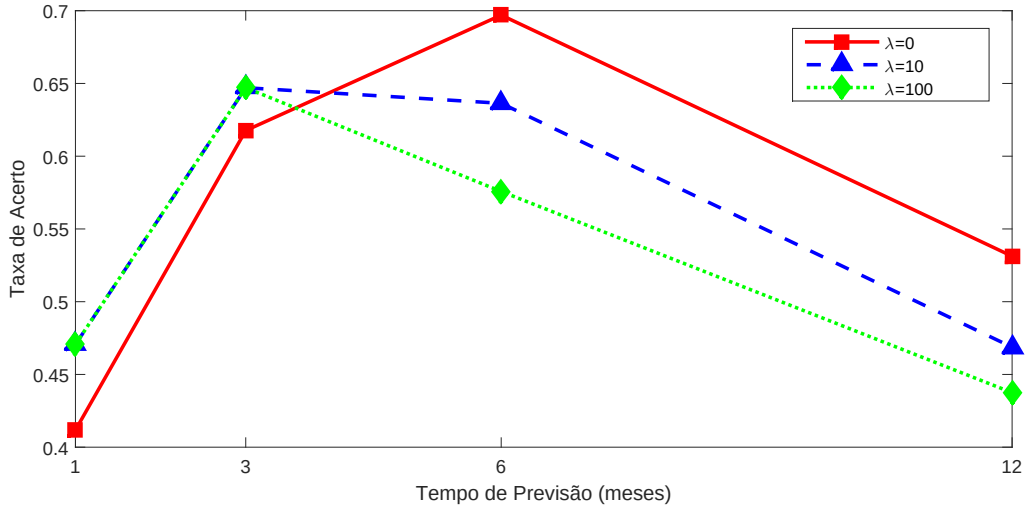


Figura 3.2: Regressão Logística.

problemas da forma

$$\begin{aligned} \min_{\theta\in\mathbb{R}^6,\xi\in\mathbb{R}^s} \quad & \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{s}\xi_i \\ s.a. \quad & y^{(i)}\left(\theta^T x^{(i)} + b\right) \geq 1 - \xi_i, \quad i = 1,\ldots,s \\ & \xi_i \geq 0, \quad i = 1,\ldots,s \end{aligned} \tag{3.3}$$

onde $y^{(i)} = (y_{treino})_i$ e $C > 0$.

Para cada $h \in \{1, 3, 6, 12\}$, o problema (3.3) correspondente foi resolvido considerando-se $C = 1$, 10^{-3}, 10^{-5} e 10^{-7}. Como (3.3) consiste na minimização de uma função quadrática convexa com restrições lineares, utilizou-se a rotina `quadprog`[1] do Matlab para resolvê-lo.

Obtido o parâmetro θ, para cada $j = 1,\ldots,|teste|$ calculou-se $\tilde{y}^{(j)}$ como

$$\tilde{y}^{(j)} = \begin{cases} 1, & \text{se } \theta^T x^{(j)} + b \geq 0 \\ -1, & \text{se } \theta^T x^{(j)} + b < 0 \end{cases},$$

com $x^{(j)}$ sendo a j-ésima linha da matriz X_{teste}, exceto primeira componente. A Tabela 3.3 contém os resultados da comparação entre $\tilde{y}$ e y_{teste}. Em particular, a Figura 3.3 apresenta um comparativo com respeito às taxas de acerto. Observa-se que as taxas de acerto das previsões para $h = 3$, 6 e 12 meses são significativamente melhores que as taxas para $h = 1$ mês. A melhor previsão ocorre para $h = 6$ meses com $C = 1$ (ou 10^{-3} ou 10^{-5}), resultando em uma taxa de acerto de quase 73%.

[1]Para mais detalhes sobre esta rotina, ver `https://www.mathworks.com/help/optim/ug/quadprog.html`

Tabela 3.3: Resultados - C-SVM

C	h	Taxa de Acerto	Falso Positivo	Falso Negativo
$C = 1$	1	0.4118	0.3529	0.2353
	3	0.6765	0.2059	0.1176
	6	**0.7273**	0.2121	0.0606
	12	0.5938	0.4063	0.0000
$C = 10^{-3}$	1	0.4118	0.3529	0.2353
	3	0.6765	0.2059	0.1176
	6	0.7273	0.2121	0.0606
	12	0.5938	0.4063	0.0000
$C = 10^{-5}$	1	0.4118	0.3529	0.2353
	3	0.6176	0.2647	0.1176
	6	0.7273	0.2121	0.0606
	12	0.5313	0.4688	0.0000
$C = 10^{-7}$	1	0.5588	0.4412	0.0000
	3	0.5882	0.4118	0.0000
	6	0.5758	0.4242	0.0000
	12	0.5000	0.5000	0.0000

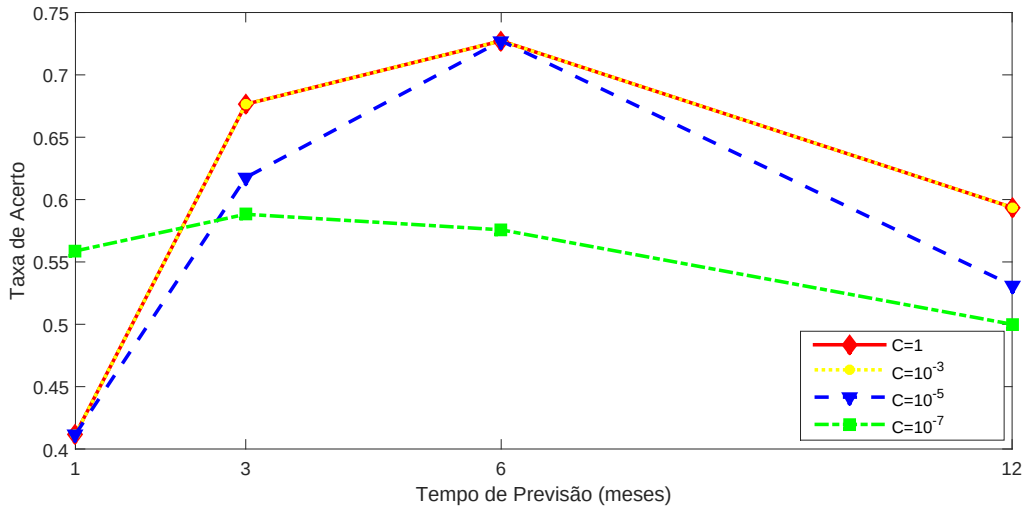


Figura 3.3: C-SVM.

3.4 Redes Neurais Artificiais

A modelagem via Redes Neurais Artificiais se deu por meio de uma rede do tipo *feed-forward*, com uma camada escondida composta de K neurônios, e tendo a função logística como função de ativação. Cada saída gerada pela rede é da forma

$$z(x) = g\left(\sum_{j=1}^{K} w'_j g(w_j^T x) + b\right). \tag{3.4}$$

Conforme descrito na Seção 2.4, a determinação dos pesos sinápticos $w'_j, b \in \mathbb{R}$ e $w_j \in \mathbb{R}^7$ sintetizados num parâmetro θ pode ser feita resolvendo-se problemas da forma

$$\min_{\theta} f(\theta) = -\sum_{i=1}^{s} \left[\bar{y}^{(i)} \log\left(z(x^{(i)})\right) + (1-\bar{y}^{(i)}) \log\left(1 - z(x^{(i)})\right)\right], \tag{3.5}$$

onde $\bar{y}^{(i)}$ é dado por (3.2).

Para cada $h \in \{1, 3, 6, 12\}$, o problema (3.5) foi resolvido considerando $K = 1$, 3, 6, 9 e 12. O método de otimização usado foi o Método quase-Newton BFGS com busca de Goldstein-Armijo (ver Capítulo 1). A matriz B_0 foi definida como sendo a matriz identidade, e as constantes da busca de Goldstein-Armijo foram definidas como $\mu_1 = 10^{-3}$ e $\mu_2 = 1 - \mu_1$. A origem foi tomada como ponto inicial e a execução do método foi interrompida no momento em que uma das condições abaixo foi satisfeita:

- $\|\nabla f(x_k)\| < 10^{-6}$
- $f(x_k) - f(x_{k+1}) < 10^{-6}$ por 50 iterações consecutivas.

Obtidos os pesos sinápticos, para cada $j = 1, \ldots, |teste|$ calculou-se

$$\tilde{y}^{(j)} = \begin{cases} 1, & \text{se } z(x^{(j)}) \geq 0.5 \\ -1, & \text{se } z(x^{(j)}) < 0.5 \end{cases},$$

com $z(x)$ dada por (3.4) e $x^{(j)}$ sendo a j-ésima linha da matriz X_{teste}. A Tabela 3.4 contém os resultados da comparação entre $\tilde{y}$ e y_{teste}. Em particular, a Figura 3.4 apresenta um comparativo com respeito às taxas de acerto. Neste caso, a melhor taxa de acerto foi de 66.67% na previsão para $h = 6$ meses, com $K = 3$ neurônios na camada intermediária.

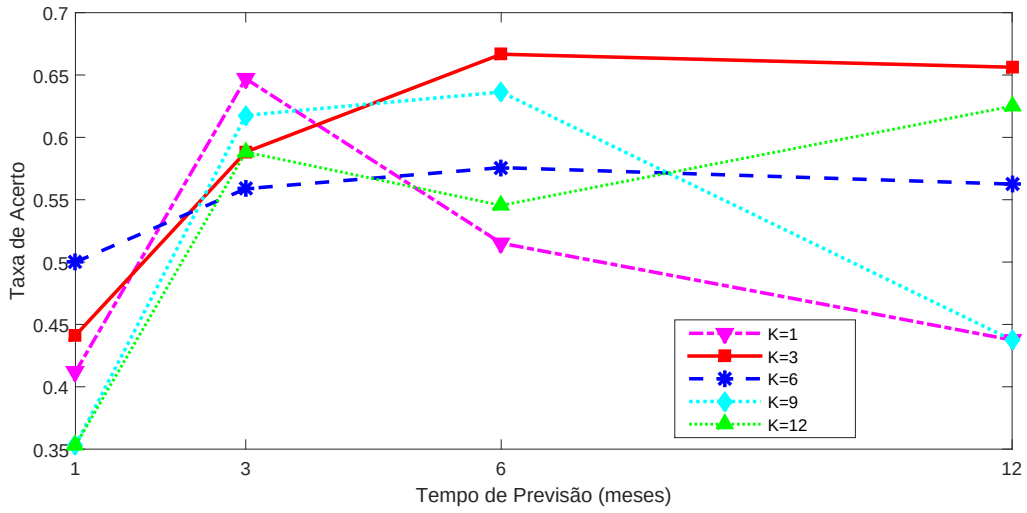


Figura 3.4: Redes Neurais.

Tabela 3.4: Resultados - Redes Neurais

K	h	**Taxa de Acerto**	**Falso Positivo**	**Falso Negativo**
1	1	0.4118	0.2647	0.3235
	3	0.6471	0.1176	0.2353
	6	0.5152	0.0303	0.4545
	12	0.4375	0.2188	0.3438
3	1	0.4412	0.2647	0.2941
	3	0.6176	0.3529	0.0294
	6	**0.6667**	0.2727	0.0606
	12	0.6563	0.1875	0.1563
6	1	0.5000	0.2059	0.2941
	3	0.5588	0.2941	0.1471
	6	0.5758	0.2727	0.1515
	12	0.5625	0.4375	0.0000
9	1	0.5588	0.4412	0.0000
	3	0.5882	0.4118	0.0000
	6	0.6061	0.3636	0.0303
	12	0.5313	0.4688	0.0000
12	1	0.3529	0.3235	0.3235
	3	0.6176	0.1765	0.2059
	6	0.6364	0.1515	0.2121
	12	0.4375	0.3125	0.2500

3.5 Regressão Linear

Conforme descrito na Seção 2.1, o modelo de Regressão Linear fornece como saída números reais, ou seja, dado um vetor $x \in \mathbb{R}^n$, o modelo devolve uma saída $m_\theta(x) = \theta^T x \in \mathbb{R}$. Portanto, a aplicação deste modelo para previsão dos movimentos do Ibovespa é feita da seguinte maneira:

- primeiro, dado um horizonte de previsão $h \in \{1, 3, 6, 12\}$, para cada $j \in [6, T-h] \cap \mathbb{N}$ busca-se prever o valor $d_{i+h} \in \mathbb{R}$, a partir do vetor

$$x^{(j)} = (1,\ d_{j-5},\ d_{j-4},\ d_{j-3},\ d_{j-2},\ d_{j-1},\ d_j)\,;$$

- então, obtido $m_\theta(x^{(j)}) = \theta^T(x^{(j)})$, calcula-se a previsão para o movimento do índice,

$$\tilde{y}^{(j)} = \text{sinal}(m_\theta(x^{(j)}) - d_j). \tag{3.6}$$

Fixado h, a determinação do parâmetro θ no modelo de Regressão Linear $m_\theta(x) = \theta^T x$ é feita resolvendo-se problemas da forma

$$\min_{\theta \in \mathbb{R}^7} f(\theta) = \|X_{treino}\theta - z_{treino}\|^2 + \lambda \|\theta\|^2, \tag{3.7}$$

onde $\lambda \geq 0$, e o vetor z_{treino} é dado por $(z_{treino})_i = d_{i+h}$, para $i = 1, \ldots, |treino|$. Para cada $h \in \{1, 3, 6, 12\}$, o problema (3.7) correspondente foi resolvido para $\lambda = 0$, 10 e 100, utilizando-se a fatoração de Cholesky (ver Seção 2.1). Obtido o parâmetro θ, para cada $j = 1, \ldots, |teste|$, calculou-se $\tilde{y}^{(j)}$ por (3.6). A Tabela 3.5 contém os resultados da comparação entre $\tilde{y}$ e y_{teste}, sendo este definido conforme descrito no início deste capítulo. Em particular, a Figura 3.5 apresenta um comparativo com respeito às taxas de acerto. Neste caso, a melhor taxa de acerto foi de 69.70% na previsão para $h = 6$ meses com $\lambda = 0$.

Tabela 3.5: Resultados - Regressão Linear

Regularização	t	**Taxa de Acerto**	**Falso Positivo**	**Falso Negativo**
$\lambda = 0$	1	0.4706	0.2647	0.2647
	3	0.6471	0.1765	0.1765
	6	**0.6970**	0.1818	0.1212
	12	0.6875	0.2813	0.0313
$\lambda = 10$	1	0.5294	0.2353	0.2353
	3	0.6176	0.1471	0.2353
	6	0.6667	0.1818	0.1515
	12	0.6875	0.2813	0.0313
$\lambda = 100$	1	0.5588	0.2059	0.2353
	3	0.6471	0.1471	0.2059
	6	0.6364	0.1818	0.1818
	12	0.6250	0.3125	0.0625

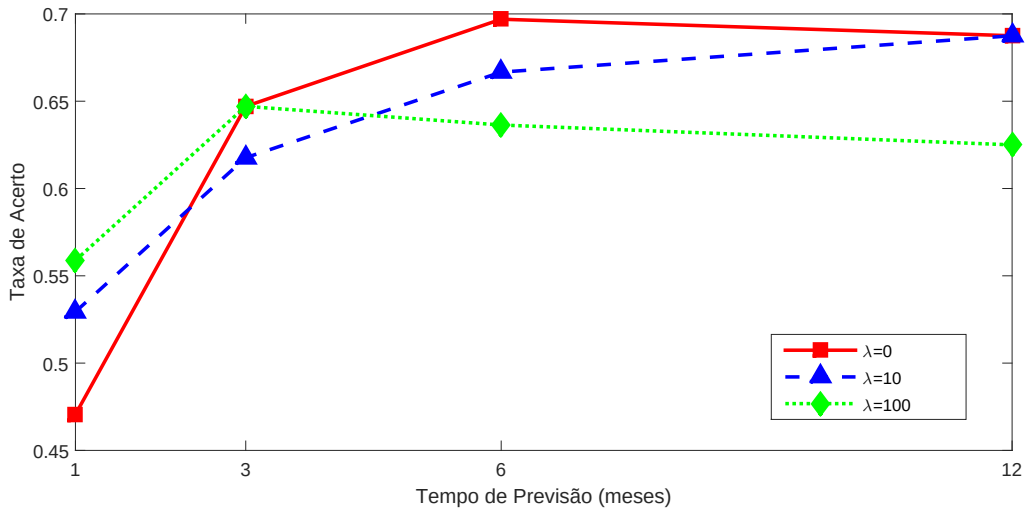


Figura 3.5: Regressão Linear.

3.6 Combinação de Modelos

De maneira geral, a previsão para 6 meses adiante apresentou os melhores resultados em termos das taxas de acerto. Para este horizonte de previsão, os melhores modelos de cada categoria apresentaram taxas superiores a 65%, conforme mostra a Tabela 3.6. Os parâmetros θ obtidos em cada um dos casos especificados nessa tabela podem ser consultados nos Anexos.

Tabela 3.6: Resultados para previsão de 6 meses

	Modelo	**Parâmetros**	**Taxa de Acerto**
m_1	Regressão Linear	$\lambda = 0$	0.6970
m_2	Regressão Logística	$\lambda = 0$	0.6970
m_3	C-SVM	$C = 1$	0.7273
m_4	Redes Neurais	$K = 3$	0.6667

A Figura 3.6 apresenta um comparativo entre os quatro modelos reportados na Tabela 3.6 para as previsões com horizonte $h = 1$, 3, 6 e 12 meses.

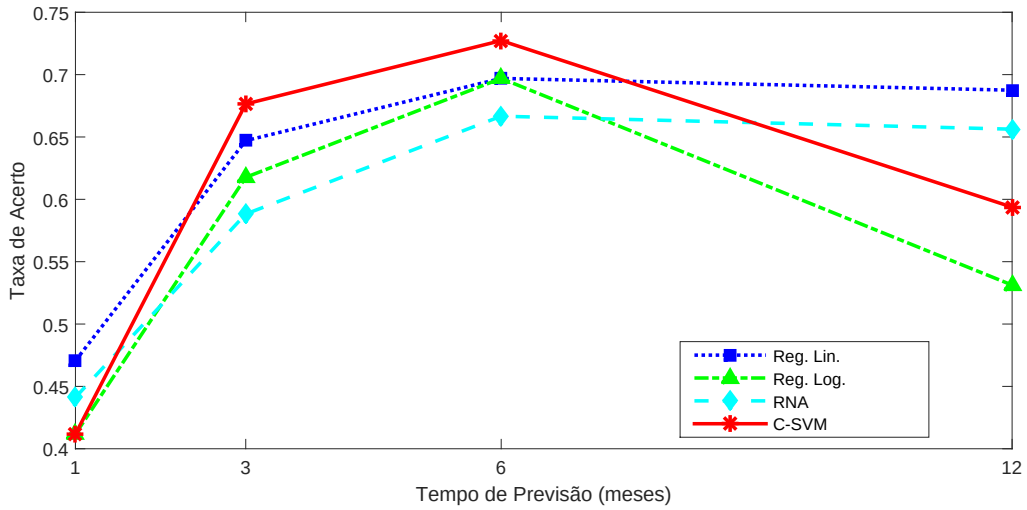


Figura 3.6: Comparação entre os modelos.

Uma estratégia com potencial para melhorar as previsões consiste na combinação dos modelos. Tal combinação pode ser feita considerando-se

$$m_c(x) = \sum_{i=1}^{k} w_i \tilde{y}_i(x),$$

onde $\tilde{y}_i(x) \in \{0, 1\}$ é a resposta do modelo i para a entrada x, k é a quantidade de modelos combinados, e w_i é o peso atribuído à saída do modelo i, com $\sum_{i=1}^{k} w_i = 1$. Note que $m_c(x) \in [0, 1]$, logo esse valor pode ser interpretado como a probabilidade da amostra

pertencer à classe positiva. Em [16], sugere-se que os pesos w_i sejam calculados por

$$w_i = \frac{TA_i}{\sum_{i=1}^{k} TA_i},$$

onde TA_i denota a taxa de acerto do modelo i. Ou seja, os maiores pesos são atribuídos aos modelos mais confiáveis. Uma vez calculado $m_c(x)$, a previsão final do modelo combinado é dada por

$$\tilde{y}_c(x) = \begin{cases} 1, & \text{se } m_c(x) \geq \eta \\ -1, & \text{se } m_c(x) < \eta, \end{cases}$$

com $\eta \in [0.5, 1)$.

A fim de se avaliar essa estratégia de combinação de modelos, foram realizados testes com todas as combinações de dois, três e quatro modelos entre aqueles reportados na Tabela 3.6, considerando-se $\eta = 0.7$. Para facilitar a apresentação dos resultados, cada modelo é denotado por um número: C-SVM (1), Regressão Logística (2), Regressão Linear (3) e Redes Neurais (4). Assim, a combinação $\{1, 2, 3\}$, por exemplo, é composta pelos modelos C-SVM, Regressão Logística e Regressão Linear. A Tabela 3.7 apresenta as taxas de acerto de cada combinação.

Tabela 3.7: Resultados - Combinação de modelos

Combinação \ Horizonte(meses)	1	3	6	12
$\{1,2,3,4\}$	0.4118	0.6176	**0.7879**	0.6563
$\{2,3\}$	**0.4412**	0.6176	**0.7879**	0.6875
$\{1,2\}$	0.4118	**0.6471**	0.7273	0.5937
$\{1,3\}$	**0.4412**	**0.6471**	**0.7879**	**0.7500**
$\{2,4\}$	0.3529	0.5882	0.7273	0.6563
$\{3,4\}$	0.3824	0.5882	0.6970	**0.7500**
$\{1,4\}$	0.3529	0.5882	0.7576	0.6563
$\{1,2,3\}$	**0.4412**	0.6176	**0.7879**	**0.7500**
$\{1,3,4\}$	0.3824	0.5882	0.7576	**0.7500**
$\{1,2,4\}$	0.3529	0.5882	0.7576	0.6563
$\{2,3,4\}$	0.3824	0.5882	0.7576	**0.7500**

Observa-se que a maior taxa de acerto foi obtida para previsão de 6 meses adiante, utilizando a combinação dos modelos C-SVM, Regressão Linear e Regressão Logística ou de todos os modelos, por exemplo. Nestes casos, a taxa de acerto chegou a 78.79%, o que representa uma melhora significativa relativamente aos modelos individuais, onde a maior taxa de acerto foi de 72.73% com o modelo C-SVM.

Já para previsão de 1 mês adiante, as combinações não superaram o desempenho dos modelos individuais. A maior taxa de acerto entre as combinações foi de 44.12%, enquanto a regressão linear chega a 47%. Algo análogo ocorre na previsão de 3 meses adiante. Enquanto o modelo C-SVM chega a 67.65% de acerto, a melhor combinação para

este caso chegou apenas a 64.71%.

Para o caso de 12 meses, também houve uma melhora significativa em relação aos modelos individuais. Utilizando a combinação de todos os modelos, uma taxa de acerto de 75% foi alcançada, enquanto o máximo obtido nos modelos individualmente foi de 68.75% (considerando os modelos com os parâmetros da Tabela 3.6).

Pela Figura 3.7, pode-se notar que, de maneira geral, o padrão observado nos modelos individuais se repete. Ou seja, a previsão de 1 mês é a menos confiável, melhora para 3 meses, atinge o auge em 6 meses e volta a decrescer na previsão de 12 meses.

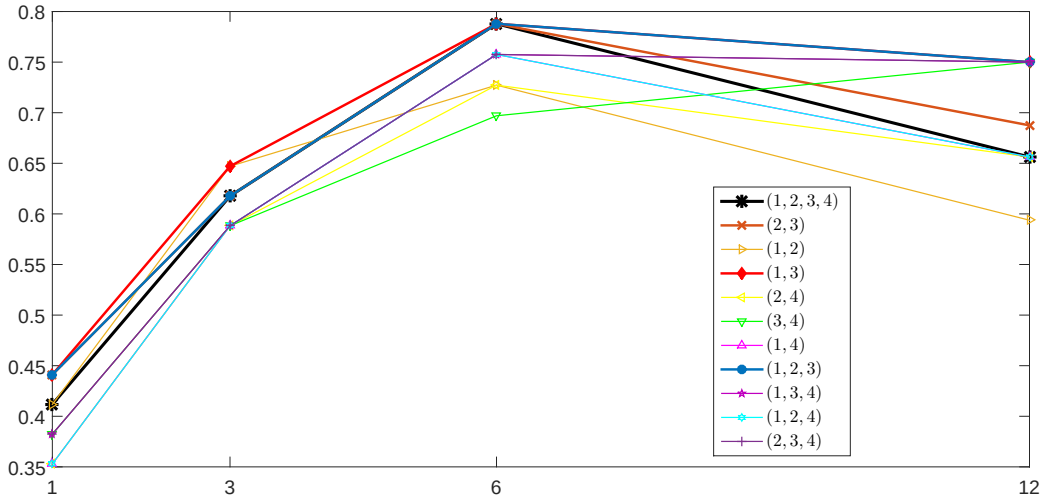


Figura 3.7: Combinação de Modelos.

Capítulo 4

Estratégia de Investimento

Este capítulo apresenta simulações para uma estratégia de investimento baseada no modelo híbrido descrito no Capítulo 3. Comparações são feitas tomando-se como referência a estratégia "buy and hold". O objeto de negociação considerado é o Fundo de Índices BOVA11, o qual acompanha os movimentos do Ibovespa com alto grau de precisão. As principais referências deste capítulo são BM&FBovespa [4], Dai e Zhang [9] e Shen et al [26].

4.1 Ibovespa e BOVA11

Apesar do foco desta dissertação ser a previsão do Ibovespa, não é possível realizar investimentos diretamente sobre este índice, ou seja, não é possível comprar e vender cotas do mesmo como se faz com ações. Uma opção para se obter rendimentos de acordo com tal índice seria negociar todas as ações que compõe a carteira teórica, obedecendo às proporções de participação de cada uma. Porém, isto seria demasiadamente trabalhoso, já que tal carteira é variável. Além disso, também poderia ser proibitivo para o pequeno investidor, tanto pelo orçamento necessário para a aquisição de todas as ações, quanto pelos custos operacionais (tais como: taxa de corretagem por ordem de compra ou venda, taxa de custódia e impostos).

Outra alternativa para replicar a performance do Ibovespa é investir no BOVA11, o qual consiste em um Fundo de Índice de Ações ou ETF (Exchange Traded Fund). Especificamente, um ETF é um investimento em renda variável, cuja negociação ocorre de maneira semelhante a uma ação. Ao adquirir uma cota de um ETF, o investidor passa a deter indiretamente todas as ações da carteira teórica do índice de referência, sem ter que comprá-las separadamente no mercado [4]. No caso do BOVA11, o índice de referência é

o Ibovespa. A Figura 4.1 mostra as séries históricas do Ibovespa e do BOVA11 no período de Janeiro de 2009 a Janeiro de 2017. Nela também pode-se observar que o BOVA11 acompanha os movimentos do Ibovespa com alto grau de precisão.

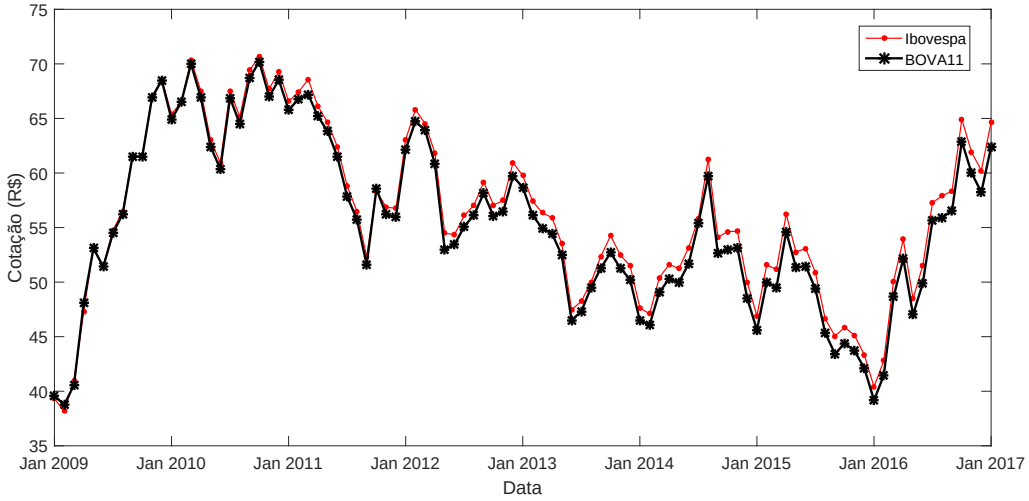


Figura 4.1: Série histórica BOVA11.

Diferente das ações usuais, cujo lote padrão é composto por 100 unidades, um lote padrão do BOVA11 é composto por apenas 10 unidades. Como as ações só podem ser negociadas em lotes, isto significa que a opção pelo BOVA11 permite investimentos mais baixos. Por exemplo, em dezembro de 2016 o BOVA11 fechou a R$ 58,24 e a ação referente ao Itaú Unibanco fechou a R$ 33,15, de modo que a negociação da primeira exigiria um investimento mínimo de R$ 582,40, enquanto na segunda seria necessário R$ 3.315,00, ou seja, um valor muito superior.

A correlação existente entre o Ibovespa e o BOVA11 sugere que uma alta no Ibovespa será acompanhada por uma alta na cotação do BOVA11, assim como uma baixa no Ibovespa será acompanhada por uma baixa na cotação do BOVA11. Portanto, previsões para o Ibovespa obtidas pelo modelo híbrido descrito no Capítulo 3 podem ser usadas para orientar a compra e venda dos lotes do BOVA11. O uso do Ibovespa nesta dissertação justifica-se pelo fato do BOVA11 ter surgido apenas no final de 2008, resultando em uma quantidade muito menor de dados para o treinamento dos modelos de Aprendizagem de Máquina.

4.2 Estratégia de Investimento

Considere um pequeno investidor hipotético que pretende negociar lotes do BOVA11 no mercado à vista. A pergunta fundamental é: de que forma esse investidor pode se bene-

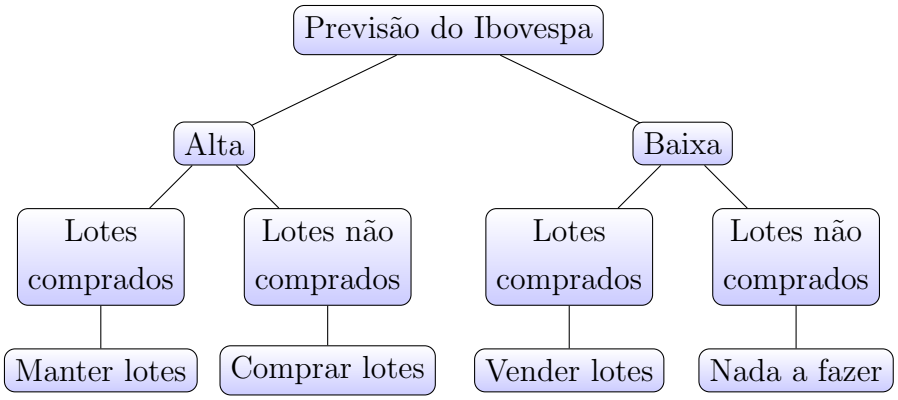
ficiar de previsões sobre os movimentos do Ibovespa? Como a melhor taxa de acerto foi obtida para o horizonte de 6 meses com o modelo híbrido (Regressão Linear, Regressão Logística e C-SVM), supõe-se que decisões de compra ou venda serão efetuadas em instantes t_0, t_1, t_2, ..., com $t_{i+1} - t_i = 6$ meses ($i = 0, 1, 2, ...$). A estratégia de investimento baseada em previsões sobre o Ibovespa consiste então nos seguintes passos:

1. Treinar os modelos de regressão linear, regressão logística e C-SVM para previsão de 6 meses adiante (ou seja, considerando $h = 6$), utilizando para treinamento os dados históricos de 2002 até t_i;

2. A partir das taxas de acerto dentro do conjunto de treinamento, construir o modelo combinado conforme descrito no Capítulo 3;

3. Aplicar o modelo combinado aos dados dos meses t_i, $t_i - 1$, $t_i - 2$, $t_i - 3$, $t_i - 4$ e $t_i - 5$, para prever se o índice estará mais alto ou mais baixo no mês t_{i+1}, ou seja, calcular $y = \text{sinal}(m_c(x) - 0.7)$, onde $x = \begin{bmatrix} t_i & t_i - 1 & t_i - 2 & t_i - 3 & t_i - 4 & t_i - 5 \end{bmatrix}$;

4. Tomar uma decisão de compra ou venda de lotes do BOVA11 de acordo com o seguinte diagrama [9]:



5. No mês t_{i+1}, defina $i = i + 1$ e volte para o passo 1.

Esta estratégia será referida como Estratégia AM, onde AM vem de Aprendizagem de Máquina. Para efeito comparativo, também será considerada a Estratégia Simples, a qual consiste em comprar o ativo no instante inicial t_0 e vendê-lo apenas no instante final t_{final} do período em estudo[1]. Supondo um investimento inicial S_0, segue abaixo uma descrição mais detalhada de cada uma dessas estratégias.

[1]Esta estratégia é também conhecida como *buy and hold*.

- **Estratégia Simples:** No instante inicial o capital todo é usado para comprar o maior número possível de lotes do BOVA11, o qual é dado por

$$q = \left\lfloor \frac{S_0 - \alpha}{10c_0} \right\rfloor,$$

onde α é a taxa de corretagem por operação e c_0 é a cotação do BOVA11 no mês de início do investimento. Assim, o saldo final é

$$S_{final} = S_0 + 10q(c_{final} - c_0) - 2\alpha,$$

onde c_{final} é a cotação do BOVA11 no último mês do período considerado.

- **Estratégia AM:** Com o saldo inicial, compra-se o máximo possível de lotes do BOVA11 no primeiro mês t_i em que o modelo de aprendizagem de máquina indicar uma valorização do Ibovespa para t_{i+1}. A partir daí, a cada 6 meses as decisões de compra e venda são reavaliadas. Quando o modelo indicar uma desvalorização do índice, todos os lotes do BOVA11 são vendidos. Então, quando o modelo indicar uma valorização do índice, todo o saldo disponível é usado para comprar o máximo de lotes possível:

$$q = \left\lfloor \frac{S_j - \alpha}{10c_j} \right\rfloor,$$

onde c_j e S_j são, respectivamente, a cotação do BOVA11 e o saldo disponível no mês t_j em questão.

Para exemplificar, suponha que $S_0 = \text{R\$ } 10.000,00$ e $\alpha = \text{R\$ } 10,00$. Considere o período de Janeiro de 2015 à Janeiro de 2016, onde tem-se as seguintes cotações do BOVA11:

$$\text{Janeiro 2015} = 45{,}61$$
$$\text{Julho 2015} = 49{,}37$$
$$\text{Janeiro 2016} = 39{,}19.$$

Na Estratégia Simples, seriam comprados 21 lotes em Janeiro de 2015, investindo um total de R\$ 9.578,10 ($21 \times 10 \times 45,61$). Em Janeiro de 2016, seriam vendidos todos estes lotes, obtendo um saldo de R\$ 8.229,90 ($21 \times 10 \times 39,19$). Assim, o saldo final, com desconto das taxas de corretagem (uma para operação de compra e outra para operação de venda), seria de

$$10.000,00 - 9.578,10 + 8.229,90 - 20,00 = 8.631,80,$$

o que representa um prejuízo de R\$ 1.368,20.

Já na Estratégia AM, há 4 casos possíveis:

- Caso 1: suponha que o modelo tenha indicado alta de Janeiro para Julho de 2015, e baixa de Julho de 2015 para Janeiro de 2016.

  Neste caso, seriam comprados 21 lotes em Janeiro de 2015, os quais seriam vendidos em Julho, obtendo um saldo final de

  $$10.000,00 - 9.578,10 + 10.367,70 - 20,00 = 10.769,60,$$

  o que representa um lucro de R$ 769,60.

- Caso 2: suponha que o modelo tenha indicado alta de Janeiro para Julho de 2015, e alta de Julho de 2015 para Janeiro de 2016.

  Neste caso, a situação seria a mesma da Estratégia Simples.

- Caso 3: suponha que o modelo tenha indicado baixa de Janeiro para Julho de 2015, e alta de Julho de 2015 para Janeiro de 2016.

  Neste caso, seriam comprados 20 lotes em Julho de 2015, os quais seriam vendidos em Janeiro de 2016. O saldo final seria dado por

  $$10.000,00 - (20 \times 10 \times 49,37) + (20 \times 10 \times 39,19) - 20,00 = 7.944,00,$$

  o que representa um prejuízo de R$ 2.056,00.

- Caso 4: suponha que o modelo tenha indicado baixa de Janeiro para Julho de 2015, e baixa de Julho de 2015 para Janeiro de 2016.

  Neste caso, o saldo inicial de R$ 10.000,00 seria mantido, pois não seria realizada nenhuma operação no período.

Outra possibilidade avaliada é o investimento de renda fixa, em particular do tipo relacionado ao CDI (Certificado de Depósito Interbancário)[2]. Primeiramente, propõe-se uma Estratégia Mista envolvendo a Estratégia AM e, nos semestres em que o dinheiro não é utilizado na compra do BOVA11, o mesmo é colocado em um investimento de renda fixa baseado no CDI. Além disso, para critério comparativo considera-se também o investimento exclusivamente em renda fixa, com rendimento vinculado ao CDI.

[2]Existem diversos tipos de investimento de renda fixa cujo rendimento é vinculado à taxa CDI, consistindo num percentual desta. Alguns exemplos de tais investimentos são CDB, LC, LCI e LCA. A porcentagem do CDI que o investimento vai render depende de muitos aspectos, dentre os quais o banco escolhido para realizá-lo. Para as simulações realizadas, considera-se um rendimento de 100% do CDI (em geral, costuma variar entre 80% e 115%).

4.3 Simulações de Investimento

Considerando $S_0 = \text{R\$ } 10.000,00$ e $\alpha = \text{R\$ } 10,00$, simulações de investimento foram realizadas para diversos períodos, a fim de se comparar o desempenho da Estratégia Simples, da Estratégia AM, da Estratégia Mista e do CDI. A Tabela 4.1 mostra os saldos finais para tais estratégias aplicadas a períodos de 1 a 8 anos, todos terminando em Janeiro de 2017.

Tabela 4.1: Simulações: períodos com término em Janeiro de 2017

Período	Estratégia Simples	Estratégia AM	Estratégia Mista	CDI
Jan2009-Jan2017	R$ 15.678,00	R$ 20.596,00	R$ 37.192,00	R$ 22.567,00
Jan2010-Jan2017	R$ 9.604,00	R$ 14.836,00	R$ 25.940,00	R$ 20.617,00
Jan2011-Jan2017	R$ 9.472,00	R$ 14.836,00	R$ 23.563,00	R$ 18.746,00
Jan2012-Jan2017	R$ 10.015,00	R$ 14.836,00	R$ 21.116,00	R$ 16.794,00
Jan2013-Jan2017	R$ 10.609,00	R$ 14.836,00	R$ 19.641,00	R$ 15.538,00
Jan2014-Jan2017	R$ 13.325,00	R$ 15.202,00	R$ 19.270,00	R$ 14.343,00
Jan2015-Jan2017	R$ 13.504,00	R$ 15.202,00	R$ 17.367,00	R$ 12.931,00
Jan2016-Jan2017	R$ 15.780,00	R$ 14.103,00	R$ 15.061,00	R$ 11.403,00

Note que a única situação em que a Estratégia Simples foi melhor que a Estratégia AM foi no período de Janeiro de 2016 à Janeiro de 2017. De maneira geral, a Estratégia AM foi significativamente melhor, especialmente nos períodos mais longos. Por exemplo, para o período de Janeiro de 2009 a Janeiro de 2017, o lucro decorrente da Estratégia AM (R$ 10.596,00) foi quase o dobro do lucro obtido com a Estratégia Simples (R$ 5.678,00). Além disso, observe que a incrementação da Estratégia AM com investimento de renda fixa nos momentos em que o dinheiro ficaria parado, ou seja, a Estratégia Mista apresentou rendimentos significativamente maiores na maioria dos casos.

As Figuras 4.2, 4.3 e 4.4 mostram a evolução do saldo para as quatro estratégias tomando como referência períodos de investimento de 7, 6 e 3 anos, respectivamente. Cada valor intermediário corresponde ao saldo que seria obtido se o investidor encerrasse o investimento naquele momento.

É possível notar que a Estratégia Mista mostrou-se bastante competitiva relativamente ao rendimento do CDI, e de modo geral claramente superior às Estratégias AM e Simples. Em relação à utilização de Aprendizagem de Máquina, pode-se observar que a Estratégia AM (assim como a Estratégia Mista) resultou em lucro ao final dos 18 períodos considerados, enquanto a Estratégia Simples resultou em prejuízo em 14 desses cenários. Mesmo nos casos em que a Estratégia Simples resultou em lucro, o lucro da Estratégia AM foi superior. Estes resultados ilustram o grande potencial do uso de técnicas de Aprendizagem de Máquina como suporte para negociações em bolsas de valores.

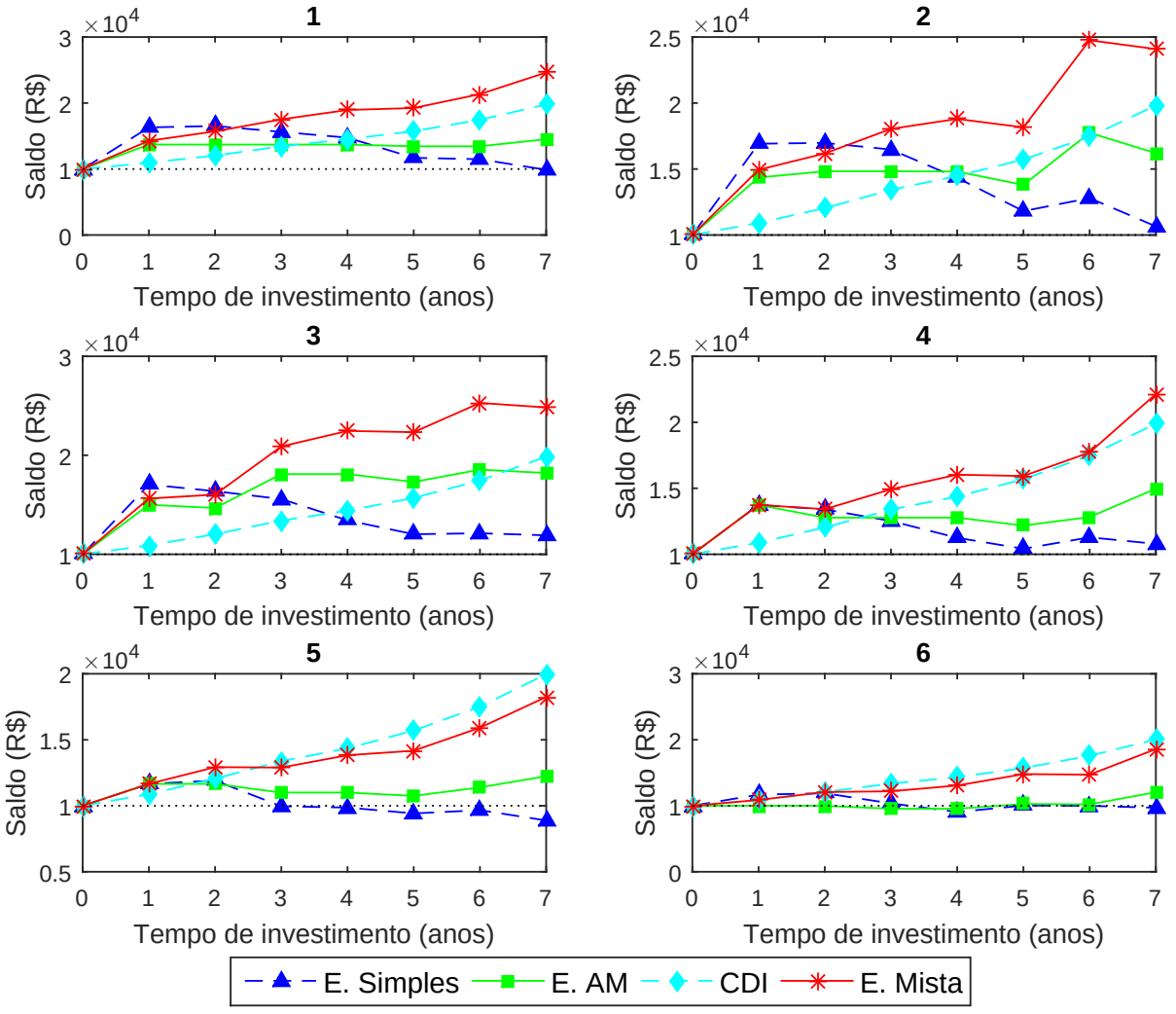


Figura 4.2: Investimentos iniciados em 2009, nos meses de Janeiro (1); Fevereiro (2); Março (3); Abril (4); Maio (5); Junho (6).

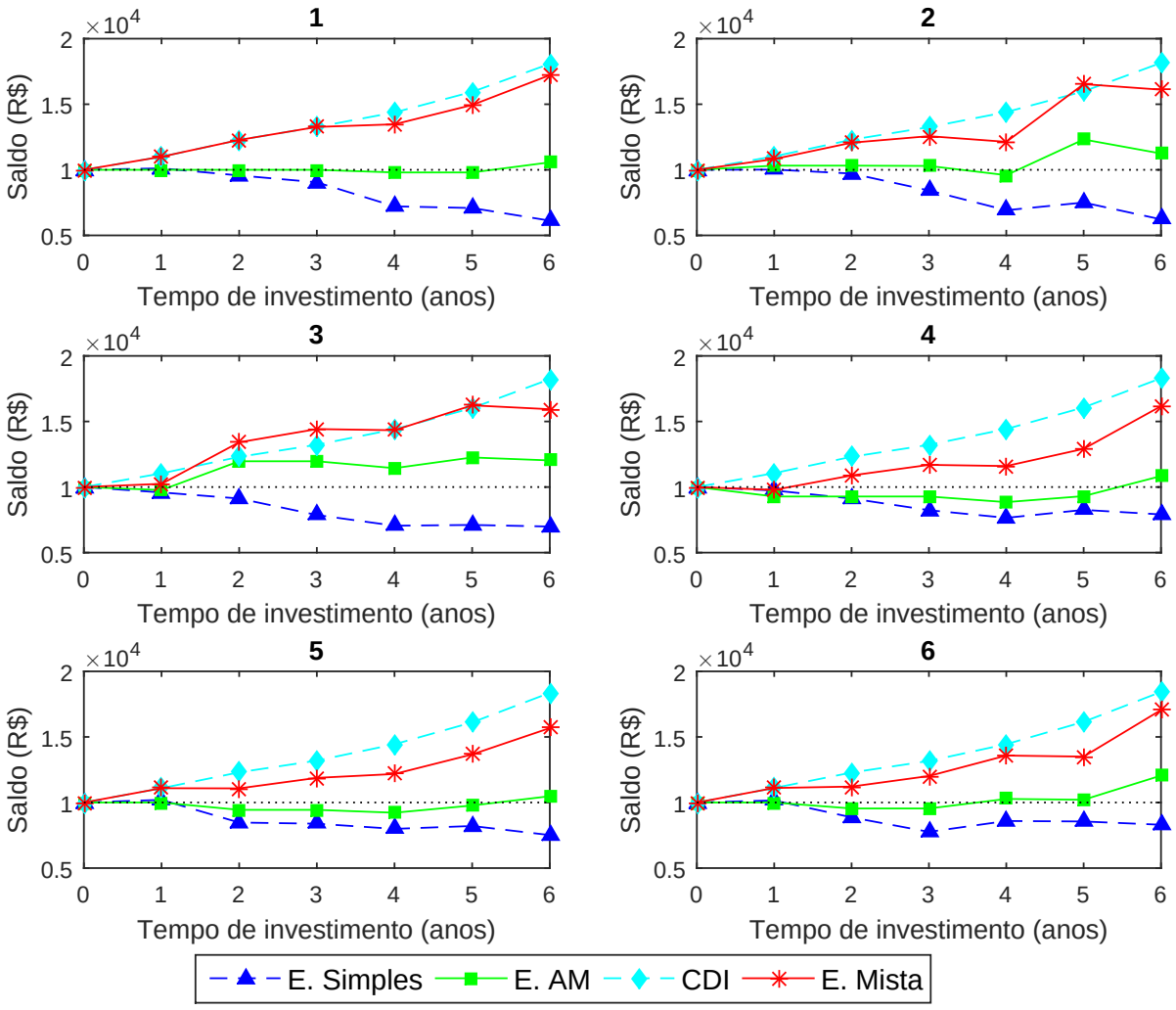


Figura 4.3: Investimentos iniciados em 2010, nos meses de Janeiro (1); Fevereiro (2); Março (3); Abril (4); Maio (5); Junho (6).

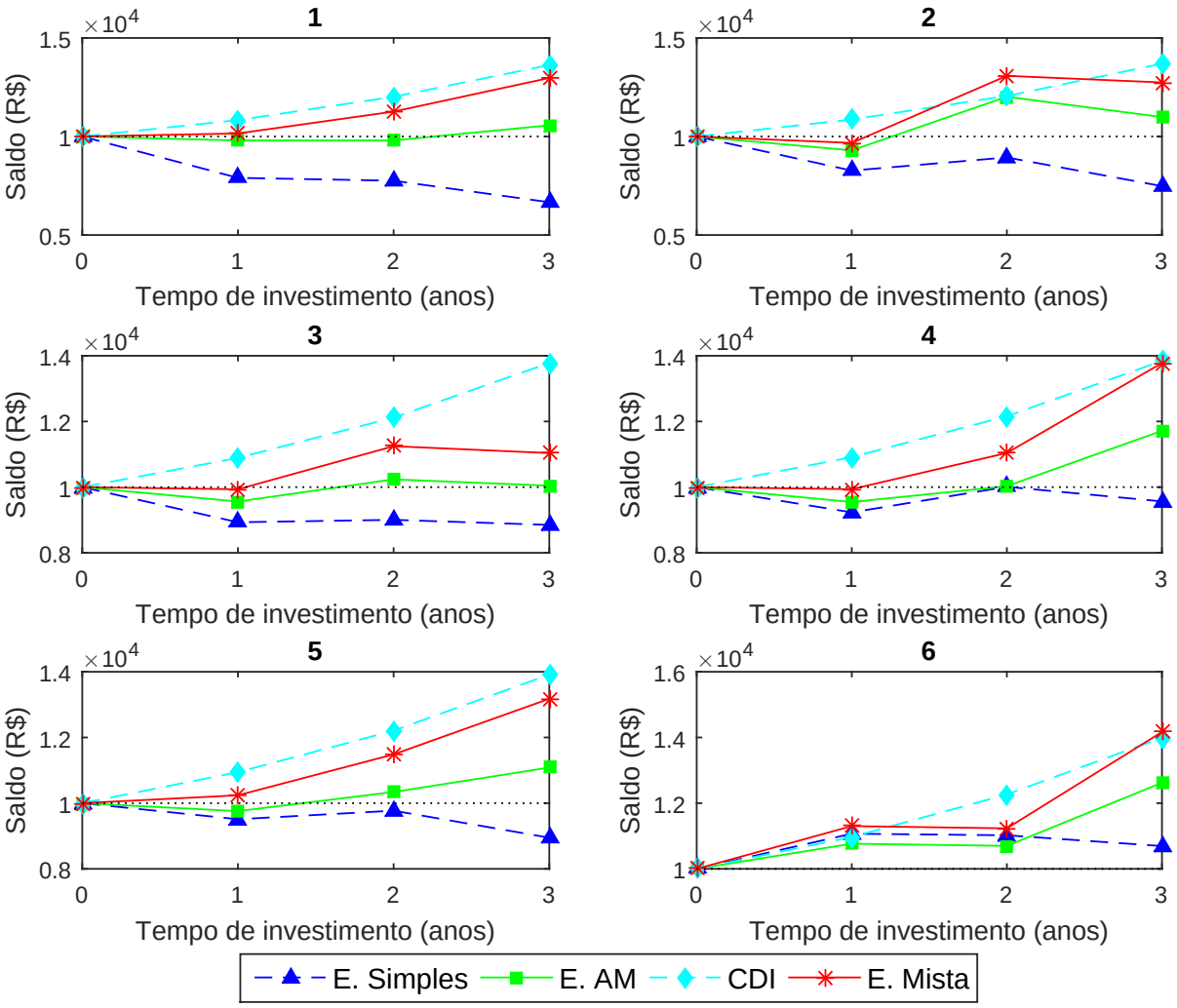


Figura 4.4: Investimentos iniciados em 2013, nos meses de Janeiro (1); Fevereiro (2); Março (3); Abril (4); Maio (5); Junho (6).

Conclusão

Neste trabalho, investigou-se o uso de técnicas de aprendizagem de máquina para a previsão dos movimentos do Ibovespa. A importância dessas previsões vem do fato de que elas podem orientar decisões envolvendo a compra e venda de ativos relacionados ao índice de referência. Matematicamente, esse problema de previsão pode ser encarado como um problema de classificação.

A busca por um classificador com boa taxa de acerto nas previsões teve como base quatro modelos de aprendizagem de máquina: Regressão Linear, Regressão Logística, Máquinas de Vetor Suporte (C-SVM) e Redes Neurais. A partir de dados mensais do Ibovespa, esses modelos foram treinados para fazer previsões binárias (de alta ou baixa do índice), com horizontes de 1, 3, 6 e 12 meses. Nos testes realizados, a melhor taxa de acerto obtida foi de 72.7% com o modelo C-SVM, para previsões com horizonte de 6 meses.

Na tentativa de se obter um modelo com uma taxa de acerto ainda maior para previsões de 6 meses, foram testadas diversas combinações dos modelos já estimados. Tais combinações resultam de médias ponderadas dos modelos, considerando-se pesos proporcionais à taxa de acerto de cada modelo. Com essa abordagem, foi possível obter uma taxa de acerto de 78.8% com uma combinação dos modelos de Regressão Linear, Regressão Logística e C-SVM.

Por fim, o melhor modelo combinado foi incorporado em uma estratégia de investimento com manutenção semestral para negociação do fundo de índices BOVA11, o qual espelha os movimentos do Ibovespa. Nas simulações, essa estratégia baseada no modelo híbrido se mostrou significativamente melhor do que a estratégia simples *buy and hold*, em termos do retorno resultante. Além disso, tal estratégia complementada com investimento em CDI mostrou-se bastante satisfatória quando comparada à aplicação exclusivamente de renda fixa. Esses resultados ilustram o grande potencial do uso de técnicas de aprendizagem de máquina como suporte para negociações em bolsas de valores, somando-se à crescente literatura sobre esse tópico [9, 16, 26, 20, 21].

O trabalho realizado nesta dissertação pode ser melhorado de várias maneiras. Alguns dos possíveis melhoramentos que serão objeto de trabalhos futuros estão descritos abaixo.

(a) Como variáveis explicativas, usou-se apenas os seis valores mais recentes do próprio Ibovespa. Assim, é razoável esperar uma melhora significativa na taxa de acerto com a inclusão de mais variáveis, tais como: a cotação do dólar, o Produto Interno Bruto (PIB) brasileiro, e o Índice Nacional de Preços ao Consumidor Amplo (IPCA).

(b) Com a inclusão de novas variáveis, técnicas de regularização se tornam extremamente importantes para evitar overfitting. Aqui, utilizou-se apenas a regularização baseada na norma l_2. Para induzir a esparsidade das soluções, é interessante considerar a regularização baseada na norma l_1, isto é, adicionar aos modelos um termo da forma $\lambda \|\cdot\|_1$. Isto resulta em problemas não suaves de otimização convexa composta, os quais podem ser resolvidos de maneira eficiente usando-se, por exemplo, métodos de primeira ordem acelerados [18, 29].

(c) No caso dos modelos C-SVM, vale a pena investigar o uso de kernels. Note que uma melhora do desempenho desse tipo de modelo pode ter um grande impacto na estratégia de investimento resultante, uma vez que o modelo sem kernel foi o modelo individual que forneceu a melhor taxa de acerto, sendo o principal componente do modelo combinado usado nas simulações.

(d) Também vale a pena testar Redes Neurais Artificiais com múltiplas camadas escondidas, apesar de isto geralmente ser mais eficiente com uma quantidade muito maior de dados. Dependendo do número de variáveis explicativas e do número de camadas escondidas, esse tipo de rede neural pode exigir a estimação de uma quantidade gigantesca de pesos sinápticos (deep learning), impondo um desafio adicional ao processo de otimização. Nesses casos, o método BFGS usado aqui poderia ser substituído por sua versão com memória limitada [18], por métodos subespaciais [29], bem como por métodos gradientes conjugados não lineares [15], por exemplo.

(e) Todos os modelos aqui utilizados exigiram a escolha de certos parâmetros, tais como a constante de regularização λ nos modelos de regressão linear e logística, a constante C no modelo C-SVM, e o número K de neurônios na camada escondida das redes neurais. Os melhores modelos em cada categoria foram obtidos testando-se algumas variações com diferentes escolhas para esses parâmetros. Uma abordagem interessante de ser investigada é a otimização da taxa de acerto desses modelos com respeito aos parâmetros. Nesse caso, para cada parâmetro ou combinação de parâmetros, a avaliação da função objetivo resultaria de uma série de simulações computacionais, tornando desafiador o cálculo de gradientes. Uma possibilidade seria então o uso de métodos de otimização sem derivadas [6].

Referências Bibliográficas

[1] A. Ng. *CS229 Lecture Notes: Supervised Learning*. Stanford University, 2016. Disponível em `http://cs229.stanford.edu/materials.html`. Acessado em: 10 de maio de 2017.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] BM&FBovespa. *Ibovespa*. `http://www.bmfbovespa.com.br/pt_br/produtos/indices/indices-amplos/indice-bovespa-ibovespa.htm`. Acessado em 4 de Abril de 2017.

[4] BM&FBovespa. *ETF de Renda Variável*. `http://www.bmfbovespa.com.br/pt_br/produtos/listados-a-vista-e-derivativos/renda-variavel/etf-de-renda-variavel.htm`

[5] B. E. Boser, I. M. Guyon, e V. N. Vapnik. A training algorithm for optimal margin classifiers. *COLT?92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY, USA: ACM Press, 144?152, 1992.

[6] A. R. Conn, K. Scheinberg e L. N. Vicente. *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization, SIAM, 2009.

[7] C. Cortes e V. Vapnik. Support-vector networks. *Machine Learning*. 20(3):273?297, 1995.

[8] G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematical Control Signals Systems*. 2: 303-314, 1989.

[9] Y. Dai e Y. Zhang. Machine Learning in Stock Price Trend Forecasting. (2013) Disponível em `http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.pdf`. Acessado em 15 de Outubro de 2016.

[10] J.E. Dennis e J. J. Moré. A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods. *Mathematics of Computation*. 28: 549-560, 1974.

[11] J.E. Dennis e J. J. Moré. Quasi-Newton Methods, Motivation and Theory. *SIAM Review*. 19:46-89, 1977.

[12] G. H. Golub e C. F. Van Loan. *Matrix Computations, 4th edition*. The John Hopkins University Press, 2013.

[13] I. Goodfellow, Y. Bengio e A. Courville. *Deep Learning*. Book in preparation for MIT Press, 2016. Disponível em `http://www.deeplearningbook.org`. Acessado em 11 de Dezembro de 2016.

[14] S. Haykin. *Neural Networks and Learning Machines, 3rd edition*. Pearson Prentice Hall, 2009.

[15] W. W. Hager e H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*. 16:170-192, 2005.

[16] W. Huang, Y. Nakamori e S. Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*. 32:2513-2522, 2005.

[17] E. L. Lima. *Análise Real - Volume 2*. IMPA, 2004.

[18] D.C. Liu e J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*. 45: 503-528, 1989.

[19] D. G. Luenberger e Y. Ye. *Linear and Nonlinear Programming*. 4 ed., Springer, 2008.

[20] S. Madge e S. Bhatt. Predicting Stock Price Direction using Support Vector Machines. Independent Work Report Spring, 2015. Disponível em `https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf`. Acessado em 15 de Outubro de 2016.

[21] M. Majumder e MD A. Hussian. Forecasting of Indian Stock Market Index using Artificial Neural Network. Disponível em `https://nseindia.com/content/research/FinalPaper206.pdf`. Acessado em 26 de Abril de 2017.

[22] K. P. Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

[23] Y. Nesterov. *Introductory Lectures on Convex Optimization - A Basic course*. Springer, 2004.

[24] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*. 27:372-376, 1983.

[25] A. A. Ribeiro e E. W. Karas. *Otimização Contínua: Aspectos Teóricos e Computacionais*. Cengage Learning, 2013.

[26] S. Shen, H. Jiang e T. Zhang. Stock Market Forecasting Using Machine Learning Algorithms. Disponível em `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.278.6139&rep=rep1&type=pdf`. Acessado em 15 de Outubro de 2016.

[27] V. N. Vapnik e A. Ja. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974.

[28] V. N. Vapnik e A. Ya. Lerner. Recognition of Patterns with help of Generalized Portraits. *Avtomat. i Telemekh.*. 24(6):774-780, 1963.

[29] ZH Wang e YX Yuan. A subspace implementation of quasi-Newton trust region methods for unconstrained optimization. *Numerische Mathematik*. 104:241-269, 2006.

[30] Yahoo Finanças. `https://br.financas.yahoo.com`. Acessado em 27 de Abril de 2016.

Anexos

Anexo 1

Valores do parâmetro θ obtido nos modelos de Regressão Linear ($\lambda = 0$), Regressão Logística ($\lambda = 0$) e C-SVM ($C = 1$)

	$h = 1$	$h = 3$	$h = 6$	$h = 12$
Regressão Linear ($\lambda = 0$)	1298.7	4458.7	9646.4	18334
	5.4395e-02	1.4419e-01	2.5213e-01	4.7022e-01
	-6.8987e-02	4.9517e-03	-6.1272e-02	-1.8617e-01
	-4.5723e-02	-2.1287e-01	-2.2100e-01	1.7749e-02
	3.9975e-02	-4.1414e-02	2.3496e-02	-1.2447e-01
	-1.6703e-01	-2.0465e-01	-1.6335e-01	-2.6082e-01
	1.1626	1.2265	0.9934	0.7581
Regressão Logística ($\lambda = 0$)	1.2704	2.7221	3.8647	16.1150
	-4.3866e-05	5.9206e-05	-1.0269e-05	-7.0124e-05
	1.1306e-04	1.4149e-05	3.2922e-05	-5.0688e-05
	-1.4302e-04	-1.1115e-05	-3.9992e-05	-8.8802e-06
	1.2111e-04	-9.2159e-05	-3.3684e-05	2.9960e-05
	-1.5694e-04	-2.9474e-05	1.0059e-04	4.1550e-05
	8.6635e-05	1.4222e-05	-1.1907e-04	-2.2355e-04
C-SVM ($C = 1$)	1.5472	2.2669	3.6088	10.183
	-4.8758e-05	8.3966e-05	1.8119e-05	-6.7727e-05
	1.5219e-04	7.9834e-06	-1.0853e-05	-1.6444e-05
	-1.7930e-04	-4.1626e-05	-1.9650e-05	1.3576e-05
	1.5210e-04	-9.4590e-05	-2.7663e-05	-3.2535e-05
	-2.5322e-04	-3.0928e-05	3.5248e-05	7.8626e-05
	1.4891e-04	3.7431e-05	-6.3074e-05	-1.6054e-04

Anexo 2

Valores do parâmetro θ obtido no modelo de Redes Neurais ($K = 3$)

	$h = 1$	$h = 3$	$h = 6$	$h = 12$
Redes Neurais ($K = 3$)	-59.8342	7041.3972	2350.0797	991.2665
	2.5818	-0.32255	-1.4188	-0.1829
	-0.96542	0.0034945	1.228	0.078717
	-2.2774	0.22651	-0.92614	-0.0048266
	0.17977	-0.37613	1.1076	-0.11846
	-0.54286	0.46315	0.67017	0.16626
	1.0625	-0.14213	-0.81371	0.023285
	1897.6227	1774.3594	5067.3994	770.9996
	0.05358	1.6255	0.17479	-0.21136
	0.65496	0.9241	0.34665	0.1037
	-0.8452	-0.84341	-0.098717	-0.27133
	0.83642	-1.402	0.18398	0.46806
	-1.8262	-0.77374	-0.10909	-0.41792
	1.073	0.3375	-0.62375	0.29076
	1129.5371	485.0429	8254.9881	754.7456
	-1.7312	-0.55272	-0.12122	0.016537
	-0.74846	0.29782	-0.1176	-0.023246
	3.0498	0.65461	-0.14707	0.020449
	-0.65468	0.057073	0.13337	-0.011571
	2.1688	-0.98392	-0.041483	0.012968
	-2.1416	0.48074	0.12301	-0.030387
	1.8611	2.9572	1.5572	3.2679
	3.9349	3.4769	2.9624	2.7775
	3.7895	1.1866	2.7632	435.0581
	-4.2615	-2.244	-2.0967	-3.3741