UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Roberto Carlos Soares Nalon Pereira Souza

Algoritmos Online Baseados em Vetores Suporte para Regressão Clássica e Ortogonal

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Raul Fonseca Neto
Coorientador: Saul de Castro Leite
Coorientador: Wagner Antônio Arbex

Juiz de Fora
2013

Roberto Carlos Soares Nalon Pereira Souza

Algoritmos Online Baseados em Vetores Suporte para Regressão Clássica e Ortogonal

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 21 de Fevereiro de 2013.

BANCA EXAMINADORA

Prof. D.Sc. Raul Fonseca Neto - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Saul de Castro Leite
Universidade Federal de Juiz de Fora

Prof. D.Sc Wagner Antônio Arbex
Empresa Brasileira de Pesquisa Agropecuária

Prof. D.Sc. Carlos Cristiano Hasenclever Borges
Universidade Federal de Juiz de Fora

Prof. Ph.D. Wagner Meira Junior
Universidade Federal de Minas Gerais

Ao meu pequeno irmão Pedro.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida e pelo sustento diário, sem o qual não teria chegado até aqui.

Agradeço a minha mãe pelo suporte incondicional, muitas vezes abrindo mão dos seus próprios objetivos para que eu concretizasse os meus. Ao meu padrasto por todo apoio e incentivo. Ao meu pequeno irmão pelos brinquedos que sempre me fazia trazer na volta dos dias de folga que tornavam a distância de casa um pouco menos solitária. À minha avó e minha tia pelas orações e preocupação constantes.

Aos meus amigos, participantes do mais diversos momentos dessa caminhada. Agradeço pela presença nos momentos de diversão e pela força nos momentos complicados. Ao Guga e ao Vidigal pela troca de experiências no decorrer do mestrado, caminho que trilhamos ao mesmo tempo, embora em instituições diferentes. Ao Iago por compartilhar a experiência de quem opta pelo mercado no lugar da academia. À Priscila e Laura pela companhia, ainda que distantes, e pelo carinho de sempre. Gostaria de agradecer ainda à muitas outras pessoas, mas o espaço aqui não me permitiria citar um por um, em especial à Karen pela companhia, ao Natan pela aventura de dividir o apartamento e à Mônica pela disposição em sempre trazer uma bagagem a mais quando de volta a Juiz de Fora.

Agradeço ao meu orientador Raul Fonseca Neto por me acolher desde os tempos da graduação, pela orientação, pelo apoio e conselhos durante o desenvolvimento deste trabalho e pela confiança no meu potencial.

Agradeço ao meu co-orientador Saul de Castro Leite por toda ajuda e apoio incansáveis dispensados à mim, os quais foram fundamentais para que este trabalho chegasse no nível em que se encontra e também pelo incentivo em continuar na pesquisa.

Ao meu co-orientador Wagner Arbex por me apoiar e incentivar desde os tempos da Embrapa, ainda antes do mestrado, principalmente quando eu estava em dúvida sobre a caminhada que vinha pela frente.

Agradeço ao amigo e professor Carlos Cristiano Hasenclever Borges, a quem considero também como co-orientador, pelo apoio desde o início do trabalho, pelas sugestões valiosas e por participar dessa banca.

Ao professor Wagner Meira pela atenção dispensada à mim quando surgi “do nada” em sua sala, em busca de uma oportunidade para continuar desenvolvendo este trabalho

e por aceitar participar dessa banca.

Agradeço aos meus amigos da turma de 2011 do mestrado em ciência da computação por compartilharem os momentos de diversão e também as dificuldades. A troca de conhecimento e experiências durante essa convivência muitas vezes foi fonte de aprendizado maior que a sala de aula.

Ao professor Guilherme Albuquerque pelas melhores aulas ministradas durante o mestrado.

Agradeço a professora Regina Braga e ao professor Marcelo Bernardes, coordenadora e vice-coordenador do mestrado, pela atenção que sempre dispensaram às minhas mais diversas solicitações como aluno. Ao professor Marcelo Bernardes agradeço ainda pelo espaço concedido à mim no Grupo de Computação Gráfica.

Aos professores do PGCC por todos os ensinamentos. Aos funcionários da secretaria/coordenação do ICE e também do DCC pelo suporte de sempre, em especial à Gláucia por toda a ajuda com as demandas do mestrado.

À todos que contribuíram para que esse objetivo se tornasse algo concreto, mas que a memória não me permitiu lembrar, os meus sinceros agradecimentos.

Por fim, mas não menos importante, agradeço à CAPES pelo apoio financeiro.

"In God we trust.

All others must have data."

William Edwards Deming

RESUMO

Neste trabalho apresenta-se uma nova formulação para regressão ortogonal. O problema é definido como a minimização do risco empírico em relação a uma função de perda com tubo desenvolvida para regressão ortogonal, chamada ρ-insensível. Um algoritmo para resolver esse problema é proposto, baseado na abordagem da descida do gradiente estocástica. Quando formulado em variáveis duais o método permite a introdução de funções kernel e flexibilidade do tubo. Até onde se sabe, este é o primeiro método que permite a introdução de kernels, através do chamado "*kernel-trick*", para regressão ortogonal. Apresenta-se ainda um algoritmo para regressão clássica que usa a função de perda ε-insensível e segue também a abordagem da descida do gradiente. Para esse algoritmo apresenta-se uma prova de convergência que garante um número finito de correções. Finalmente, introduz-se uma estratégia incremental que pode ser usada acoplada com ambos os algoritmos para obter soluções esparsas e também uma aproximação para o "tubo mínimo" que contém os dados. Experimentos numéricos são apresentados e os resultados comparados a outros métodos da literatura.

Palavras-chave: Regressão Ortogonal. Métodos Kernel. Algoritmos Online. Máquinas de Vetores Suporte.

ABSTRACT

In this work, we introduce a new formulation for orthogonal regression. The problem is defined as minimization of the empirical risk with respect to a tube loss function developed for orthogonal regression, named ρ-insensitive. The method is constructed via an stochastic gradient descent approach. The algorithm can be used in primal or in dual variables. The latter formulation allows the introduction of kernels and soft margins. To the best of our knowledge, this is the first method that allows the introduction of kernels via the so-called "kernel-trick" for orthogonal regression. Also, we present an algorithm to solve the classical regression problem using the ε-insensitive loss function. A convergence proof that guarantees a finite number of updates is presented for this algorithm. In addition, an incremental strategy algorithm is introduced, which can be used to find sparse solutions and also an approximation to the "minimal tube" containing the data. Numerical experiments are shown and the results compared with other methods.

Keywords: Orthogonal Regression. Kernel Methods. Online Algorithms. Support Vector Machines.

LISTA DE FIGURAS

2.1 Mínimos Quadrados Ordinário × Mínimos Quadrados Total 18
2.2 Função de perda ε-insensível. 19
2.3 Função de perda ρ-insensível. 20

7.1 Processo do algoritmo de estratégia incremental. 47

8.1 Tratando variáveis simetricamente. 51
8.2 Introduzindo regularização. 57
8.3 Relação entre os pontos de treinamento e targets para os conjuntos de dados gerados. 59

LISTA DE TABELAS

8.1 Informações sobre as bases de dados. 52
8.2 Resultados obtidos pela regressão ortogonal (ρPRF) e clássica (εPRF e SVM-light), comparando esparsidade e qualidade da solução sob diferentes intensidades de ruído. 53
8.3 Resultados da regressão ortogonal (ρPRF) e clássica (εPRF e SVM-light) sem permitir flexibilidade na margem para obter uma aproximação para o tubo mínimo contendo os dados. 55
8.4 Resultados obtidos na execução do ρPRF_{AES} e ρPRF_{AES}-reg com 1000 e 5000 iterações para a base de dados *Sinc*. Comparações com o SVM-light são apresentadas 56
8.5 Informações sobre as bases de dados. 58
8.6 Resultados comparando tempo de execução em grandes bases de dados entre o $\varepsilon\text{PRF}_{AES}$ e o SVM-light. 60
8.7 Informações sobre as bases de dados 60
8.8 Informações sobre as bases de dados 62

SUMÁRIO

1 INTRODUÇÃO 13

1.1 MOTIVAÇÃO E TRABALHOS CORRELATOS 14

1.2 OBJETIVOS 15

1.3 ORGANIZAÇÃO 16

2 O PROBLEMA DE REGRESSÃO E FUNÇÕES DE PERDA 17

3 MÉTODOS TRADICIONAIS 22

3.1 MíNIMOS QUADRADOS VIA EQUAÇÕES NORMAIS 22

3.2 MíNIMOS QUADRADOS VIA DVS 23

3.3 MíNIMOS QUADRADOS TOTAL 24

3.4 REGRESSÃO NÃO LINEAR 26

4 MÉTODOS KERNEL 28

4.1 FUNDAMENTAÇÃO TEÓRICA 28

4.2 REGRESSÃO BASEADA EM VETORES SUPORTE 30

4.2.1 Formulação Primal 31

4.2.2 Formulação Dual 31

5 ALGORITMOS ONLINE PARA REGRESSÃO 34

5.1 PERCEPTRON DE ε-RAIO FIXO 34

5.1.1 Prova de Convergência 35

5.2 PERCEPTRON DE ρ-RAIO FIXO 37

6 ALGORITMO DUAL 39

6.1 εPRF DUAL 40

6.2 ρPRF DUAL 40

6.3 OTIMIZAÇÕES COMPUTACIONAIS 41

6.4 FLEXIBILIDADE 42

6.5 REGULARIZAÇÃO 43

7 ESTRATÉGIA INCREMENTAL 45
7.1 ORDENANDO OS DADOS 48

8 EXPERIMENTOS 50
8.1 TRATANDO VARIÁVEIS SIMETRICAMENTE 50
8.2 ESPARSIDADE 51
8.3 TUBO MíNIMO 54
8.4 INTRODUZINDO REGULARIZAÇÃO 55
8.5 TEMPO DE EXECUÇÃO 57
8.6 *BENCHMARK* 60

9 CONSIDERAÇÕES FINAIS 63
9.1 TRABALHOS FUTUROS 63

REFERÊNCIAS 65

1 INTRODUÇÃO

O problema de regressão consiste em encontrar uma relação desconhecida entre determinados pontos $x_i \in \mathbb{R}^n$ e seus correspondentes valores observados (geralmente chamados de *targets*) $y_i \in \mathbb{R}$. Esse problema normalmente é formulado como o de encontrar uma função $f : \mathbb{R}^n \to \mathbb{R}$, que mapeia os pontos aos valores observados, minimizando determinada função de perda. No caso da regressão clássica, assume-se que o ruído está presente apenas nos valores observados e a função de perda mede os desvios de $f(x_i)$ para os correspondentes y_i. Esse é o caso da formulação proposta por Gauss, que minimiza a soma dos erros ao quadrado entre os valores observados e a função estimada (HUBER, 1972). Esse método, chamado de mínimos quadrados encontra a melhor estimativa dentro do princípio de minimização do risco empírico, quando os ruídos relativos a amostragem real dos dados são gerados identicamente segundo uma distribuição normal (RAWLINGS *et al.*, 1998).

A aplicação para problemas de regressão encontra um vasto campo na comunidade científica. Dentre as principais áreas pode-se citar física, economia, biologia, medicina, processamento de sinais, engenharias, entre outros, em que é comum realizar determinado experimento a partir do qual diversas variáveis são medidas e deseja-se mapear o fenômeno descrito por esses dados. Em geral, esse tipo de problema exige um mapeamento não linear e os métodos tradicionais (como no caso do método dos mínimos quadrados), embora possam ser estendidos para a solução desse tipo de problema, tendem a obter resultados que não são satisfatórios. Uma abordagem que apresenta bons resultados para problemas não lineares são os métodos baseados em kernel (SMOLA e SCHÖLKOPF, 2002) cujo representante mais importante são as máquinas de vetor suporte (SVM).

Para resolver o problema da regressão clássica, (VAPNIK, 1995) desenvolveu uma formulação baseada em uma função de perda chamada ε-insensível e introduziu o conceito de tubo. Esses novos elementos, baseados no princípio de minimização do risco estrutural, permitiram o desenvolvimento de uma formulação de máquina de vetores suporte específica para problemas de regressão, chamada regressão-SV (SVR). Esse método tornou-se bastante popular devido a sua flexibilidade, especialmente em relação ao uso de funções kernel (SMOLA e SCHÖLKOPF, 2002). O conceito de tubo permite a representação

da solução final somente em termos dos vetores suporte, o que é crucial para métodos baseados em kernel. Em sua formulação padrão, a regressão-SV requer a solução de um problema de otimização quadrática, que demanda alto custo computacional, principalmente para problemas em larga escala e no caso de aplicações que dependem do tempo.

A regressão ortogonal, por outro lado, tem suas origens com (ADCOCK, 1877) (veja (MARKOVSKY e HUFFEL, 2007) para uma revisão histórica). Esse problema de regressão aparece na literatura sob diferentes nomes, por exemplo, ele foi chamado de mínimos quadrados total (GOLUB, 1973; GOLUB e LOAN, 1980) e é comumente chamado de erro-nas-variáveis na comunidade estatística (MARKOVSKY e HUFFEL, 2007; GRILICHES e RINGSTAD, 1970). Nesse contexto, o ruído pode se apresentar não somente nos valores observados y_i, mas também nos pontos x_i.

Atualmente, esse problema é motivado por inúmeras aplicações, como, por exemplo, em processamento de aúdio (HERMUS *et al.*, 2005) e imagens (LUONG *et al.*, 2012, 2011; HIRAKAWA e PARKS, 2006), visão computacional (MÜHLICH e MESTERLKOPF, 1998), astronomia (BRANHAM, 1995) e quimiometria (SCHUERMANS *et al.*, 2005) (veja (MARKOVSKY, 2010) para uma lista mais completa). A abordagem usual para resolver o problema de regressão ortogonal é através do método de decomposição em valores singulares da matriz dos dados (MARKOVSKY e HUFFEL, 2007).

1.1 MOTIVAÇÃO E TRABALHOS CORRELATOS

Como discutido anteriormente os métodos tradicionais, em geral, obtém resultados não satisfatórios quando estendidos para problemas não lineares. Assim os métodos baseados em kernel, em especial a regressão-SV, surgem como uma abordagem eficaz para essa classe de problemas. Contudo, no caso da SVR, a necessidade de solução de um problema de otimização quadrática com restrição demanda um longo tempo de processamento, principalmente para problemas em larga escala. Dessa maneira, torna-se importante a construção de novas soluções, no sentido de evitar o elevado custo computacional desse método. Além disso, métodos eficientes podem ser a chave para diversos tipos de aplicação, como no caso de problemas em que a solução completa não se faz necessária, no entanto, deseja-se obter uma aproximação de maneira mais rápida, ou ainda para o caso em que a solução de diversos problemas de regressão estão embutidos no desfecho de um problema maior e portanto devem ser computados de forma eficiente.

No contexto de problemas de classificação, em que a função estimada assume valores discretos, uma atenção considerável tem sido empenhada no desenvolvimento de algoritmos simples e eficientes, para construir classificadores de larga margem que evitam a complexidade da programação quadrática. Alguns exemplos da vasta literatura incluem (SUYKENS e VANDEWALLE, 1999; GENTILE, 2001; LI e LONG, 2002; KIVINEN *et al.*, 2004; SHALEV-SHWARTZ *et al.*, 2007; LEITE e NETO, 2008).

Em (KIVINEN *et al.*, 2004) um grupo de algoritmos online, chamados em conjunto de NORMA, é introduzido. Dentre eles um algoritmo para regressão clássica é apresentado. O problema de regressão é brevemente discutido e um algoritmo é derivado usando uma versão modificada da função de perda ε-insensível, embora não sejam fornecidos dados numéricos do método proposto. A ideia desse algoritmo é adaptar o raio do tubo ε à medida que itera através dos dados. Esse processo, entretanto, pode resultar em uma solução que não é esparsa, uma vez que diversos pontos no conjunto de treinamento podem contribuir para a solução final. Uma abordagem semelhante é apresentada em (CRAMMER *et al.*, 2006), em que um conjunto de algoritmos para diferentes tarefas de predição são desenvolvidos, incluindo o problema de regressão clássica considerando a perda ε-insensível.

(BI e BENNET, 2003) propõem uma interpretação geométrica do problema de regressão clássica, transformando-o em um problema de classificação binária para um dado valor de ε. A princípio, pode-se usar essa técnica para estender algoritmos de classificação para problemas de regressão clássica. Contudo, esse procedimento produz um conjunto duplicado de pontos, tornando-o assim, menos atrativo para aplicações práticas.

Em relação à regressão ortogonal, embora o interesse atual da comunidade científica seja crescente, principalmente no que diz respeito a aplicações, a literatura ainda carece do desenvolvimento de métodos online para a solução desse tipo de problema.

1.2 OBJETIVOS

O objetivo principal desse trabalho é o estudo e desenvolvimento de métodos online para regressão.

Para regressão clássica é apresentado um algoritmo que usa a função de perda ε-insensível e segue a abordagem da descida do gradiente. Ideias semelhantes para esse algoritmo já foram propostas na literatura, contudo uma nova prova de convergência é

apresentada neste trabalho, que garante um número finito de correções.

Em relação à regressão ortogonal apresenta-se uma nova formulação que adapta a ideia da função de perda ε-insensível. O problema é definido como a minimização do risco empírico em relação a uma nova função de perda com tubo desenvolvida para regressão ortogonal, chamada de ρ-insensível. Um algoritmo para resolver esse problema é proposto, baseado na abordagem da descida do gradiente estocástica, similar ao Perceptron (ROSENBLATT, 1958). O método proposto pode ser usado na forma primal ou dual, tornando-o mais flexível para diferentes tipos de problemas. Em sua formulação dual, o algoritmo permite a introdução de funções kernel e flexibilidade do tubo. Até onde se sabe, este é o primeiro método que permite a introdução de kernels, através do chamado "*kernel-trick*", para regressão ortogonal.

Além disso, uma estratégia incremental, que pode ser usada em conjunto com os métodos de regressão clássica e ortogonal é introduzida. Essa estratégia pode ser usada para obter soluções mais esparsas e também uma aproximação para o "tubo mínimo" que contém os dados.

1.3 ORGANIZAÇÃO

O trabalho está estruturado como a seguir. A seção 2 introduz formalmente o problema de regressão e diferentes funções de perdas. Na seção 3 são apresentados os métodos tradicionais de regressão clássica e ortogonal. A seção 4 revisa a teoria de métodos kernel e descreve a formulação SVR. Na seção 5, o framework para algoritmos online é apresentado e os algoritmos propostos no trabalho são derivados em variáveis primais. Isso constitui a base do método proposto. Na seção 6 os algoritmos são desenvolvidos em variáveis duais. A seção 7 introduz a estratégia incremental para encontrar soluções esparsas e também uma aproximação para o tubo mínimo quem contém os dados. Além disso, experimentos numéricos e resultados são reportados na seção 8 para suportar a teoria. Finalmente, a seção 9 apresenta algumas conclusões e discussões.

2 O PROBLEMA DE REGRESSÃO E FUNÇÕES DE PERDA

Seja $X_m := \{x_i\}_{i=1}^m$, com $x_i \in \mathbb{R}^n$, o conjunto de pontos de treinamento e $Y_m := \{y_i\}_{i=1}^m$, com $y_i \in \mathbb{R}$ os correspondentes valores observados (ou *targets*). Seja $Z_m := \{(y, x) : y \in Y_m \text{ e } x \in X_m\}$ o conjunto de treinamento. O problema geral de regressão é definido como: suponha que os pares $z_i := (y_i, x_i) \in Z_m$ são amostras independentes de um vetor aleatório $Z := (Y, X)$, em que Y e X são correlacionados e possuem uma distribuição conjunta desconhecida $\mathcal{P}_Z$. Dado Z_m, o problema é encontrar uma relação desconhecida entre os pontos e seus respectivos valores observados, dada pela função $f : \mathbb{R}^n \to \mathbb{R}$, sobre uma determinada classe $\mathcal{C}$ de funções, que minimiza o *risco esperado*:

$$\mathrm{E}_Z[\ell(Y, X, f)],$$

em que a esperança é tomada em relação à distribuição $\mathcal{P}_Z$ e $\ell : \mathbb{R} \times \mathbb{R}^n \times \mathcal{C} \to \mathbb{R}$ é a *função de perda*, que penaliza os desvios entre o funcional e os valores observados.

Uma abordagem nesse caso é usar Z_m para estimar $\mathcal{P}_Z$, entretanto, isso geralmente se torna uma tarefa mais desafiadora do que o problema original. Por isso, é comum considerar o problema de encontrar uma função $f \in \mathcal{C}$ que minimiza o *risco empírico* dado o conjunto de treinamento Z_m, isto é:

$$R_{\text{emp}}[f, Z_m] := \frac{1}{m} \sum_{i=1}^{m} \ell(y_i, x_i, f).$$

Com o interesse de aplicar o chamado "*kernel trick*" posteriormente, restringe-se a classe de funções $\mathcal{C}$ à funções lineares na forma: $f_{(w,b)}(x) := \langle w, x \rangle + b$, em que $w \in \mathbb{R}^n$ é o vetor de pesos e $b \in \mathbb{R}$ é o bias.

A escolha mais comum para ℓ é a perda quadrática dada por:

$$\ell_2(y, x, f) := (y - f(x))^2,$$

que dá origem ao método de mínimos quadrados. A lógica por trás dessa abordagem é minimizar a soma dos resíduos ao quadrado $\delta y_i := y_i - f(x_i)$ de tal maneira que

$y_i = f(x_i) + \delta y_i$. A suposição comum é que apenas os valores observados possuem ruído. Entretanto, alguns problemas práticos podem apresentar ruído nos pontos de treinamento x_i. Uma generalização natural do processo anterior é também minimizar variações nos pontos x_i, ou seja, minimizar $\delta y_i^2 + \delta x_i^2$ de tal forma que $y_i = f(x_i + \delta x_i) + \delta y_i$. Esse processo é comumente chamado de mínimos quadrados total ou regressão ortogonal (MARKOVSKY e HUFFEL, 2007). Geometricamente esse problema minimiza a soma das distâncias ortogonais ao quadrado entre os pontos $z_i := (y_i, x_i)$ e o hiperplano

$$\{(y, x) \in \mathbb{R}^{n+1} : y - \langle x, w \rangle = b\} \equiv \{z \in \mathbb{R}^{n+1} : \langle z, (1, -w) \rangle = b\},$$

ao contrário da formulação de mínimos quadrados, que minimiza a soma das diferenças diretas ao quadrado entre os valores funcionais e os observados, como visto na figura 2.1.

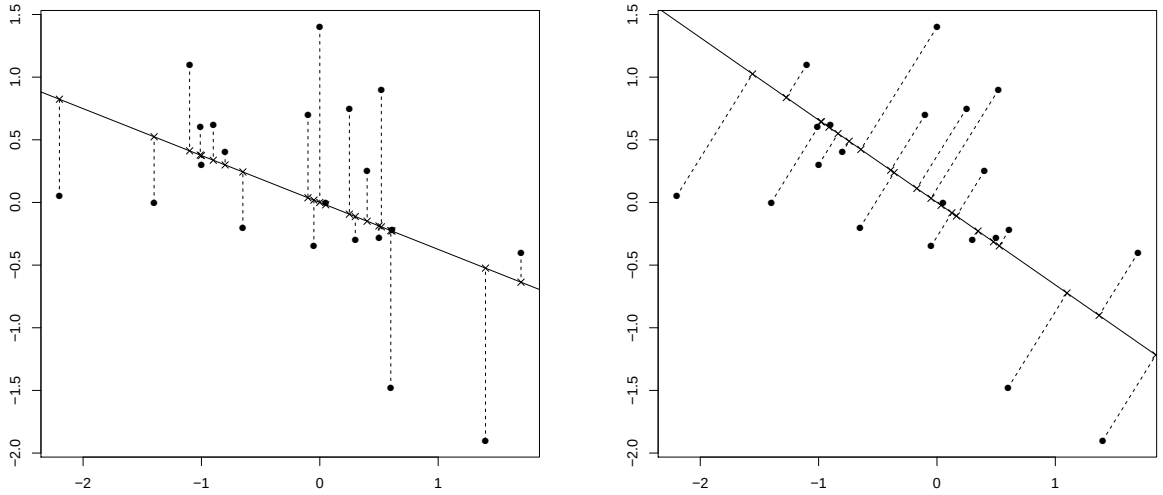


Figura 2.1: *esquerda*: Ajuste por mínimos quadrados ordinário. *direita*: Ajuste por mínimos quadrados total.

Para enquadrar o problema de mínimos quadrados total no framework introduzido no começo da seção, inicia-se definindo a p-distância de um ponto $z \in \mathbb{R}^n$ para o hiperplano $\mathrm{H} := \{z \in \mathbb{R}^n : \langle z, w \rangle = b\}$ como $\mathrm{dist}_p(z, \mathrm{H}) := \min_{x \in \mathrm{H}} ||z - x||_p$, em que $||x||_p$ é a p-norma do vetor x. É possível escrever essa distância como:

$$dist_p(z, \mathrm{H}) = \frac{\langle z, w \rangle + b}{||w||_q},$$

em que $||\cdot||_q$ é a norma conjugada de $||\cdot||_p$, com $1/p + 1/q = 1$, veja por exemplo (DAX, 2006). Assim, a função de perda correspondente para a formulação de mínimos quadrados

total pode ser escrita como:

$$\ell_t(y, x, f_{(w,b)}) := \frac{(y - \langle w, x\rangle - b)^2}{||(1, w)||^2},$$

em que a norma $||\cdot||$ sem índice inferior corresponde a norma L_2, $||\cdot||_2$.

Outra escolha comum para função de perda, que é usada na regressão-SV, é chamada de perda ε-insensível (ou ε-tubo), dada por:

$$\ell_\varepsilon(y, x, f) := \max\{0, |y - f(x)| - \varepsilon\},$$

em que ε é tomado como o raio desse tubo. A interpretação dessa função de perda pode ser feita da seguinte forma: caso o ponto esteja posicionado dentro do tubo, não há perda e o valor do resíduo não é considerado na função de erro. Entretanto, caso o ponto esteja situado fora do tubo, a perda é dada pela quantidade $|y - f(x)| - \varepsilon$. A figura 2.2 descreve essa interpretação da função de perda ε-insensível.

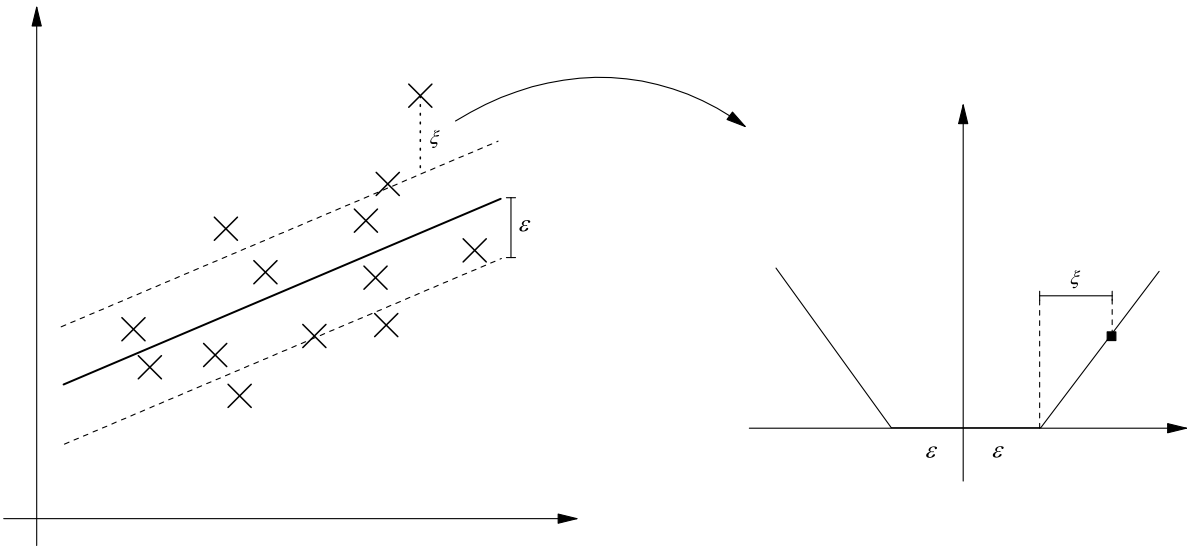


Figura 2.2: Função de perda ε-insensível.

Uma característica favorável dessa função é que ela fornece *soluções esparsas* quando o problema é formulado em variáveis duais. Em relação a essa função de perda, algumas notações e terminologias que serão usadas posteriormente são introduzidas a seguir. Para

cada $\varepsilon > 0$ fixo, define-se o seguinte conjunto:

$$\mathcal{V}(Z_m, \varepsilon) := \{(w, b) \in \mathbb{R}^{n+1} : |y_i - \langle w, x_i \rangle - b| \leq \varepsilon, \forall (x_i, y_i) \in Z_m\},$$

chamado de *espaço de versões*. Quando esse conjunto é não vazio, considera-se que o problema aceita um tubo de tamanho ε, ou um ε-tubo.

No sentido de considerar uma função de perda para problemas de regressão ortogonal, que seja útil para manter a esparsidade da solução dual, propõe-se a seguinte função: para um $\rho > 0$, seja

$$\ell_\rho(y, x, f_{(w,b)}) := \max\left\{0, \frac{|y - \langle w, x \rangle - b|}{||(1, w)||} - \rho\right\},$$

que recebe o nome de perda ρ-insensível (ou ρ-tubo). Dessa forma, a função de perda penaliza soluções que deixam pontos do lado de fora desse tubo considerando a distância ortogonal. A figura 2.3 ilustra a função de perda ρ-insensível. De maneira similar ao que

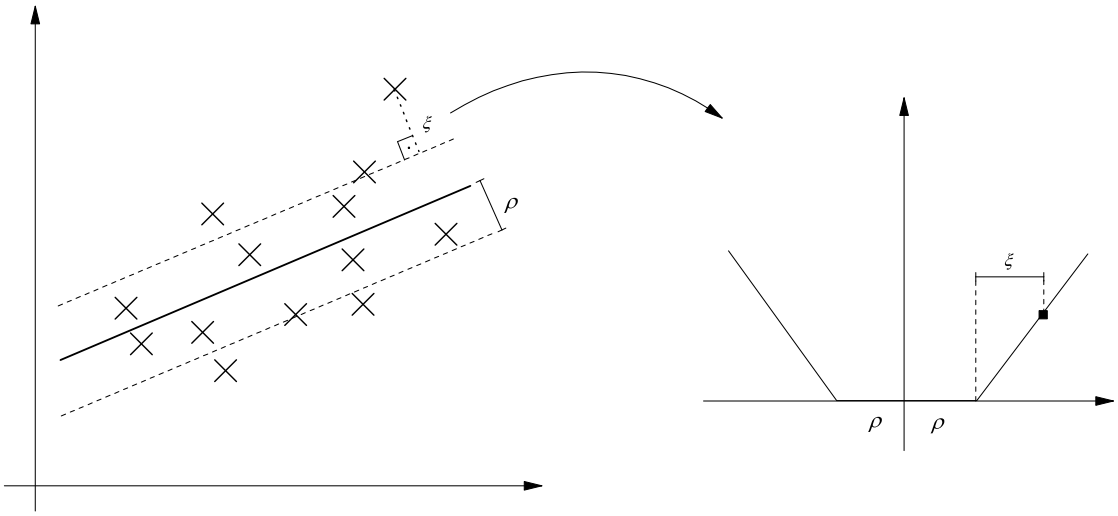


Figura 2.3: Função de perda ρ-insensível.

foi feito para a função ℓ_ε, define-se o seguinte espaço de versões:

$$\Omega(Z_m, \rho) := \{(w, b) \in \mathbb{R}^{n+1} : |y_i - \langle w, x_i \rangle - b| \leq \rho||(1, w)||, \forall (x_i, y_i) \in Z_m\},$$

para cada $\rho > 0$. Novamente, considera-se que o problema aceita um ρ-tubo se o espaço

de versões é não vazio.

Para cada (w, b) fixo, observe que existe um relação interessante entre as distâncias ortogonais e as diferenças funcionais diretas para cada ponto (y_i, x_i). Para isso, seja $\varepsilon_i := y_i - \langle w, x_i \rangle - b$ e $\rho_i := (y_i - \langle w, x_i \rangle - b)/||(1, w)||$, então claramente $\varepsilon_i = \rho_i ||(1, w)||$.

3 MÉTODOS TRADICIONAIS

No contexto da análise numérica (veja por exemplo (WATKINS, 2002)), o problema de regressão é geralmente apresentado como a seguir: Seja $X \in \mathbb{R}^{m \times n}$ e $Y \in \mathbb{R}^m$ os dados do problema, coletados, por exemplo, a partir de algum experimento. Deseja-se encontrar $w \in \mathbb{R}^n$ tal que $Xw \approx Y$. Se $m = n$ e X possui inversa, então é possível resolver $Xw = Y$. Contudo, geralmente o que se tem é um sistema super-determinado em que $m > n$ e portanto um número infinito de soluções podem ser obtidas.

Uma solução para esse problema é considerar um vetor de resíduos $r = y - Xw$. A solução é então obtida tomando w de tal maneira que a norma do vetor resíduo $||r||$, seja a menor possível. Se a norma escolhida for a norma Euclidiana, essa solução é equivalente à minimização da função ℓ_2 apresentada na seção anterior, que dá origem ao método dos mínimos quadrados.

3.1 MÍNIMOS QUADRADOS VIA EQUAÇÕES NORMAIS

O problema agora consiste em resolver o sistema super-determinado na forma

$$Xw = Y,$$

em que $X \in \mathbb{R}^{m \times n}$ e $Y \in \mathbb{R}$. Assim, a solução do problema é obtida através da minimização da norma Euclidiana dos resíduos ao quadrado:

$$\min_{w \in \mathbb{R}^n} ||Y - Xw||^2.$$

Assume-se que a matriz X possui posto completo, i.e. posto$(X) = n$. Então, a solução é obtida observando os pontos críticos da função descrita pelo resíduo:

$$\begin{aligned} r(w) = ||Y - Xw||^2 &= (Y - Xw)^T(Y - Xw) \\ &= Y^TY - Y^TXw - w^TX^TY + w^TX^TXw. \end{aligned}$$

Tomando o gradiente dessa função e igualando a zero, leva à:

$$\nabla r(w) = 0 \Rightarrow -2X^T Y + 2X^T Xw = 0$$
$$\Leftrightarrow X^T Xw = X^T Y.$$

Esse sistema recebe o nome de *equações normais*. Vale destacar que a solução de mínimos quadrados é de interesse especial quando o ruído associado ao vetor Y segue uma distribuição normal com média zero e variância σ^2 (WATKINS, 2002). A solução das equações normais pode ser obtida através da decomposição de Cholesky (FILHO, 2007).

É importante mencionar que a construção das equações normais na solução de mínimos quadrados muitas vezes não é uma boa escolha, já que o número de condicionamento da matriz $X^T X$ é o quadrado da matriz X, o que pode levar a problemas numéricos. A próxima seção apresenta uma maneira de obter a solução de mínimos quadrados sem a necessidade da construção das equações normais.

3.2 MÍNIMOS QUADRADOS VIA DVS

A Decomposição em Valores Singulares (DVS) (STRANG, 1993; GOLUB e LOAN, 1996) consiste em fatorar uma matriz de dados $X \in \mathbb{R}^{mxn}$ de tal maneira que:

$$X = U\Sigma V^T,$$

em que $U \in \mathbb{R}^{mxm}$ e $V \in \mathbb{R}^{nxn}$ são matrizes ortogonais, e $\Sigma \in \mathbb{R}^{mxn}$ é uma matriz diagonal na forma:

$$\begin{bmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{bmatrix},$$

em que $\hat{\Sigma} \in \mathbb{R}^{nxn}$.

Considerando ainda o problema de regressão da seção anterior tem-se o seguinte sistema super-determinado:

$$Xw = Y.$$

A solução dos mínimos quadrados considera então a minimização dos resíduos:

$$\| Y - Xw \|^2 = \| y - U\Sigma V^T w \|^2,$$

pela decomposição em valores singulares. Como U é ortogonal vale que $U^T = U^{-1}$ e o valor da norma não é alterado, então:

$$\| Y - Xw \|^2 = \| U^T Y - U^T U\Sigma V^T w \|^2 = \| U^T Y - \Sigma V^T w \|^2 .$$

Fazendo $U^T Y = a$ e $V^T w = b$ com

$$a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad e \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

em que $a_1, b_1 \in \mathbb{R}^n$ e $a_2, b_2 \in \mathbb{R}^{m-n}$. Então

$$\begin{aligned} \| Y - Xw \|^2 = \| U^T Y - \Sigma V^T w \|^2 = & \left\| \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} - \begin{bmatrix} \hat{\Sigma} b_1 & 0 \\ 0 & 0 \end{bmatrix} \right\|^2 \\ = & \| a_1 - \hat{\Sigma} b_1 \|^2 + \| a_2 \|^2 . \end{aligned}$$

Assim, a soma dos desvios ao quadrado será mínima quando b_1 for a solução do sistema diagonal $\hat{\Sigma} b_1 = a_1$. Como $V^T w = b$, e V é ortogonal, tem-se a solução

$$w = V b_1.$$

A solução dos mínimos quadrados usando a decomposição em valores singulares fornece uma maneira de manter a estabilidade numérica. Contudo, a complexidade computacional e a quantidade de memória necessária é maior do que a solução via equações normais com decomposição de Cholesky (FILHO, 2007) e a melhor escolha de algoritmo pode variar de acordo como problema.

3.3 MÍNIMOS QUADRADOS TOTAL

No contexto da regressão ortogonal uma generalização natural para o método dos mínimos quadrados foi introduzida por Golub & Van Loan (1973; 1980) e recebeu o nome de mínimos quadrados total (MQT). Na solução anterior usando mínimos quadrados, a

suposição comum é que somente os valores observados estão sujeitos a ruído, de tal maneira que $Y = Xw + r$, em que $r \in \mathbb{R}^m$. No caso do método de mínimos quadrados total a matriz de dados X é considerada também sujeita a erros de modo que $(Y+r) = (X+E)w$, com $E \in \mathbb{R}^{m \times n}$. Assim, seguindo (GOLUB e LOAN, 1996), considera-se o problema de regressão da seguinte maneira:

$$\min_{w \in \mathbb{R}^n} ||E, r||^2.$$

A abordagem usual para resolver o problema de regressão ortogonal através do método dos mínimos quadrados total consiste em aplicar a decomposição em valores singulares à matriz de dados (MARKOVSKY e HUFFEL, 2007). Comumente, considera-se que o problema de maneira mais geral, em que pode-se ter múltiplos targets, ou seja $Y \in \mathbb{R}^{m \times d}$. Nesse caso, a solução pode ser obtida da seguinte forma:

Seja $Z := [XY]$, com $X \in \mathbb{R}^{m \times n}$ e $Y \in \mathbb{R}^{m \times d}$, a matriz de dados associados ao problema. Pela decomposição em valores singulares tem-se que $Z = U\Sigma V^T$, com $\Sigma = diag(\sigma_1, \ldots, \sigma_{n+d})$. Além disso, a matriz V é particionada da seguinte forma:

$$V := \begin{array}{c} \begin{array}{cc} n & d \end{array} \\ \left[\begin{array}{cc} V_{11} & V_{12} \\ V_{21} & V_{22} \end{array} \right] \begin{array}{c} n \\ d \end{array} \end{array}$$

A solução MQT existe se, e somente se, V_{22} é não singular (MARKOVSKY e HUFFEL, 2007). Além disso, com $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{n+d}$, essa solução é única se, e somente se, $\sigma_n \neq \sigma_{n+1}$. Para o caso em que é única, a solução pode ser obtida por:

$$w = -V_{12}V_{22}^{-1}.$$

Para o caso mais comum em que $d = 1$, como normalmente aparece o problema de regressão, tem-se $Y \in \mathbb{R}^m$. Assim, define-se $\alpha := V_{22}$ e tem-se que $\alpha \in \mathbb{R}$. Dessa forma, caso $\alpha = 0$, o problema MQT não possui solução (GOLUB e LOAN, 1996). Em caso contrário a solução é dada por:

$$w = -V_{12}/\alpha.$$

É importante observar a interpretação geométrica do método de mínimos quadrados total, como destacado em (GOLUB e LOAN, 1996). É possível mostrar que a solução MQT é equivalente a minimizar

$$\psi(w) := \frac{||Y - Xw||^2}{||w||^2 + 1},$$

que corresponde a distância ortogonal entre a superfície da solução e os pontos de treinamento. Note que esta função é equivalente à função de perda ℓ_t apresentada na seção 2.

A maior parte dos problemas de MQT na prática podem ser resolvidos dessa maneira (MARKOVSKY e HUFFEL, 2007). Detalhes relativos ao método de mínimos quadrados total, como por exemplo casos em que a solução não é única, são tratados em (HUFFEL e VANDEWALLE, 1991). Além disso, modificações e extensões, como por exemplo MQT para problemas de grande porte, são apresentados em (HUFFEL, 1997) e (HUFFEL e LEMMERLING, 2002).

3.4 REGRESSÃO NÃO LINEAR

Até então foi discutido apenas o caso em que deseja-se ajustar uma função linear aos dados de entrada. Contudo, de maneira geral os problemas encontrados não costumam ter esse comportamento linear. Para esses casos, o modelo pode ser naturalmente generalizado de forma a obter uma curva não linear que melhor descreve os dados, minimizando também a perda quadrática. Nesse caso, uma opção é ajustar por exemplo um polinômio de grau. Isso leva à representação da função que se deseja obter, de funções lineares na forma $f(x) = \langle w, x \rangle + b$, para um polinômio

$$f(x) = w_1 x + w_2 x^2 + \ldots + b.$$

A solução pode ser obtida de maneira similar a que foi apresentada na seção 3 através da construção das equações normais. Vale destacar ainda, que a obtenção de funções não lineares não se limita apenas a funções polinomiais. De fato, qualquer tipo de curva pode ser construída escolhendo-se uma base e procedendo da mesma maneira descrita anteriormente. O desafio consiste em escolher adequadamente essa função. A seção seguinte apresenta uma solução elegante para obtenção de funções não lineares sem a necessidade

de escolher uma base adequada para a função.

4 MÉTODOS KERNEL

Os métodos kernel constituem uma classe de algoritmos de reconhecimento de padrões e aproximação de funções que usam uma função kernel como sua medida de similaridade. A teoria que fundamenta a construção desses métodos é atribuída ao trabalho de (ARONSZAJN, 1950). Essa teoria permitiu o desenvolvimento de máquinas com capacidade de resolver problemas não lineares de maneira simples, mapeando o problema em um espaço de mais alta dimensão em que a solução pode ser mais facilmente obtida. Sua utilização foi proposta inicialmente por (AIZERMAN *et al.*, 1964) para solução de problemas de reconhecimento de padrões, introduzindo funções kernel no algoritmo do perceptron para aplicação em problemas não linearmente separáveis. Contudo, somente nos anos 90, com o surgimento das Máquinas de Vetores Suporte (BOSER *et al.*, 1992) esse estudo ganhou popularidade e atenção da comunidade científica.

Esta seção faz uma breve introdução aos fundamentos teóricos de aprendizado com kernel e apresenta a formulação clássica da regressão baseada em vetores suporte proposta por (VAPNIK, 1995).

4.1 FUNDAMENTAÇÃO TEÓRICA

Um conjunto não vazio $\mathcal{X}$ de elementos (e.g, vetores) forma um *espaço linear* se a esse conjunto estão associadas duas operações: adição de elementos em $\mathcal{X}$, e o produto entre elementos de $\mathcal{X}$ e números reais, guardando determinadas propriedades (AKHIEZER e GLAZMAN, 1993). Se esse espaço linear $\mathcal{X}$ possui produto interno $\langle \cdot, \cdot \rangle \, \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$:, então $\mathcal{X}$ recebe o nome de *espaço pré-Hilbert* (HUNTER e NACHTERGAELE, 2001).

Definição 4.1.1. (AKHIEZER e GLAZMAN, 1993) Um *espaço de Hilbert* $\mathcal{H}$, é um espaço equipado com produto interno $\langle \cdot, \cdot \rangle$ e que é completo em relação à métrica gerada pelo produto interno. Observe que a métrica em $\mathcal{H}$ corresponde à norma, que pode ser naturalmente definida pelo produto interno como $|| \cdot || = \sqrt{\langle \cdot, \cdot \rangle}$.

Definição 4.1.2. (SMOLA e SCHÖLKOPF, 2002) Seja $\mathcal{X}$ um conjunto não vazio e $\mathcal{H}$ um espaço de Hilbert de funções $f : \mathcal{X} \rightarrow \mathbb{R}$. Então, $\mathcal{H}$ é um *espaço de Hilbert reproduzido por kernel* (RKHS), se existe uma função $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ com as seguintes propriedades:

1. $\forall x \in \mathcal{X}$ a função $k(x, \cdot)$ pertence a $\mathcal{H}$;

2. Propriedade reprodutiva: $\forall f \in \mathcal{H}$ e $\forall x \in \mathcal{X}$, $f(x) = \langle f, k(x, \cdot)\rangle$.

Um dos resultados matemáticos mais importantes que fundamenta a teoria do aprendizado com kernels foi apresentado por Mercer (MERCER, 1909; ARONSZAJN, 1950). Informalmente, esse resultado permite verificar se determinada função $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ é um kernel, correspondendo portanto a um produto interno em determinado espaço. Naturalmente, essa função k deve ser simétrica, condição derivada da simetria do produto interno. Além disso, a matriz $K \in \mathbb{R}^{m \times m}$, chamada de matriz kernel, com componentes $K_{ij} := k(x_i, x_j)$, para $x_i, x_j \in X_m$, deve ser positiva semi-definida (SMOLA e SCHÖLKOPF, 2002). Dessa forma, na prática basta escolher uma função k que atende as condições de Mercer e é possível mostrar que existe um RKHS para o qual k é o kernel associado.

Alguns exemplos de funções kernel comumente utilizados são:

- Linear: $k(x_i, x_j) = \langle x_i, x_j\rangle$.

- Polinomial: $k(x_i, x_j) = (\langle x_i, x_j\rangle + 1)^d$, $d \in \mathbb{N}$.

- Gaussiano: $k(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$, $\gamma > 0$.

Observe ainda que deseja-se resolver o problema em um espaço de mais alta dimensão, possivelmente infinita como no caso do kernel Gaussiano. Portanto, é importante verificar como representar computacionalmente uma função com um número infinito de parâmetros com uma quantidade finita de memória. Esse resultado é dado pelo teorema da representação, apresentado a seguir.

Teorema 4.1.1. (KIMELDORF e WAHBA, 1971; SMOLA e SCHÖLKOPF, 2002) Denota-se por $\omega : [0, \infty) \to \mathbb{R}$ uma função monótona estritamente crescente, por $\mathcal{X}$ um conjunto e por $\ell : \mathcal{X} \times \mathbb{R}^2 \to \mathbb{R} \cup \{\infty\}$ uma função de perda arbitrária. Então, cada função $f \in \mathcal{H}$ que minimiza funcional de risco regularizado

$$\sum_{i=1}^{m} \ell(y_i, x_i, f_i) + \omega(||f||^2_{\mathcal{H}}),$$

admite uma representação da forma

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x).$$

Assim, ainda que deseja-se resolver o problema de otimização em um espaço $\mathcal{H}$ de dimensão infinita, a solução recai na expansão dos m pontos do conjunto de treinamento que é finito, garantido a representação de f como um vetor em $\mathbb{R}^m$.

4.2 REGRESSÃO BASEADA EM VETORES SUPORTE

O primeiro algoritmo a introduzir a abordagem de vetores suporte foi proposto por (VAPNIK e LERNER, 1963) e as máquinas de vetores suporte (SVM) no formato atual são uma generalização desse primeiro método. Originalmente desenvolvidas para resolver problemas de reconhecimento de padrões (BOSER *et al.*, 1992), as SVM foram posteriormente estendidas para o problema geral de aproximação de funções (VAPNIK, 1995).

Um dos aspectos fundamentais da solução SVM é a possibilidade de representar a superfície de decisão em termos de um pequeno subconjunto dos dados, chamados vetores suporte (SMOLA e SCHÖLKOPF, 2002). Com o objetivo de manter a característica do método também no contexto de aproximação de funções, Vapnik introduziu a função de perda ε-insensível e o conceito de tubo. Esses novos elementos, tornaram possíveis a aplicação de vetores suporte ao problema de regressão, permitindo assim, dentro do princípio de minimização do risco estrutural, o desenvolvimento de uma máquina de vetores suporte específica para esses problemas, denominada *Support Vector Regression* ou regressão-SV.

A introdução da abordagem de vetores suportes ao problema de regressão é feita através da determinação de um tubo de raio ε, fixado a priori, que deverá conter todos os pontos do conjunto de treinamento. Esse valor representa a máxima perda admitida para cada ponto do conjunto.

A compensação entre a complexidade do modelo e a minimização dos erros residuais está relacionada à introdução de um conjunto de variáveis de folga ξ_i e ξ_i^* que flexibilizam, a exemplo da margem flexível na SVM (CORTES e VAPNIK, 1995), a pertinência dos pontos a região delimitada pelo tubo.

Assim, dado o conjunto de treinamento Z_m, a função $f(x) = \langle w, x \rangle + b$, é obtida

através da minimização do seguinte risco regularizado (TIKHONOV e ARSENIN, 1977)):

$$R_{\text{reg}} := \frac{1}{2}||w||^2 + C\ell_\varepsilon(y, x, f),$$

em que $\ell_\varepsilon(y, x, f)$ representa o risco empírico associado à função de perda ε-insensível apresentada na seção 2. A minimização da norma quadrática $||w||^2$ é considerada no sentido de reduzir a complexidade do função estimada com vista a uma maior generalização, produzindo uma função com a propriedade de *flatness* (SMOLA e SCHÖLKOPF, 1998).

4.2.1 FORMULAÇÃO PRIMAL

Para assegurar a viabilidade primal do problema introduz-se, para um dado raio ε, as seguintes restrições associadas ao conjunto de variáveis de folga, ξ_i e ξ_i^*, para cada ponto do conjunto de dados.

$$\begin{gathered} y_i - \langle w, x_i\rangle - b \leq \varepsilon + \xi_i, \text{para } \langle w, x_i\rangle + b \leq y_i \\ \langle w, x_i\rangle + b - y \leq \varepsilon + \xi_i^*, \text{para } \langle w, x_i\rangle + b \geq y_i \\ \xi_i, \xi_i^* \geq 0 \end{gathered}$$

Introduzindo as restrições de relaxação relacionadas à formação do tubo, obtém-se, o seguinte problema de otimização quadrática restrita para a regressão-SV em sua forma primal, estabelecido por (VAPNIK, 1995):

$$\text{minimizar } \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

$$\text{sujeito a: } \begin{cases} y_i - \langle w, x_i\rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{4.1}$$

A constante $C > 0$ pondera o compromisso entre a complexidade do modelo e a quantidade de desvios maiores que ε admitidos (SMOLA e SCHÖLKOPF, 1998).

4.2.2 FORMULAÇÃO DUAL

A formulação dual fornece o meio de estender a regressão-SV para funções não lineares, através do uso de funções kernel. Esta formulação é obtida construindo a função La-

grangiana a partir da função objetivo do problema em sua forma primal e as respectivas restrições, introduzindo um conjunto de variáveis duais:

$$\begin{aligned} L := \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) - \sum_{i=1}^{m}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ - \sum_{i=1}^{m}\alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b) \\ - \sum_{i=1}^{m}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b), \end{aligned} \tag{4.2}$$

Elimina-se então as variáveis primais do problema, observando as condições de primeira ordem associadas às derivadas parciais da função Lagrangiana. Essas derivadas quando substituídas na equação (4.2), levam ao seguinte problema de otimização em sua forma dual, adotando a introdução de funções kernel para obtenção de funções não lineares. Assim, o problema em sua forma dual, seguindo (SMOLA e SCHÖLKOPF, 1998) corresponde a:

$$\begin{aligned} &\text{maximizar} \begin{cases} \frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\kappa(x_i, x_j) \\ -\varepsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \end{cases} \\ &\text{sujeito a:} \begin{cases} \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned} \tag{4.3}$$

O vetor w pode ser reescrito como $w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i$, portanto $f(x) = \sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)\kappa(x_i, x_j)$ b. O termo bias pode ser computado através das condições de Karush-Kuhn-Tucker (KKT) (KUHN e TUCKER, 1951):

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b) = 0, \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b) = 0, \\ (C - \alpha_i)\xi_i = 0, \\ (C - \alpha_i^*)\xi_i^* = 0. \end{aligned}$$

Convém observar que, a exemplo do classificador SVM com margem flexível (CORTES e VAPNIK, 1995), os pontos para os quais $\alpha_i = C$ ou $\alpha_i^* = C$, são aqueles que se encontram

fora da região delimitada pelo tubo. Os pontos em que $0 < \alpha_i \leq C$ ou $0 < \alpha_i^* \leq C$ são os vetores suportes, isto é, o subconjunto de pontos que descreve a função estimada. Os pontos em que $\alpha_i = 0$ e $\alpha_i^* = 0$ não interferem na construção do hiperplano. Além disso, pode-se concluir que $\alpha_i \alpha_i^* = 0$, ou seja, ambas variáveis simultaneamente não podem ter seus valores diferentes de zero, o que implica em dizer que ambas as inequações não podem estar ativas ao mesmo tempo na solução ótima do problema

É importante mencionar que existem várias extensões para a formulação padrão do problema de regressão-SV. Essas extensões incluem a parametrização do raio do tubo (SCHÖLKOPF *et al.*, 1998) e a utilização de diferentes funções de perda (VAPNIK, 1995; CAMPS-VALLS *et al.*, 2006).

Finalmente, vale destacar que a regressão-SV é um método altamente confiável e tem demonstrado sucesso nas aplicações em diversos problemas reais nas mais diversas áreas. Contudo, a solução em *batch* do problema de programação quadrática necessária, muitas vezes demanda grande tempo de processamento. Além disso, esse tempo está diretamente relacionado ao tamanho do conjunto de treinamento. Nesse sentido, a abordagem *on-line* surge como uma solução, principalmente para problemas com grande quantidade de dados.

5 ALGORITMOS ONLINE PARA REGRESSÃO

No contexto de aprendizado *online* a função candidata $f \in \mathcal{C}$ (geralmente chamada de hipótese) é construída através da minimização do risco empírico examinando um exemplo de treinamento (y_i, x_i) por vez. Dessa maneira, inicia-se com uma hipótese inicial f_0 e, a cada iteração t, o algoritmo examina um exemplo e atualiza a hipótese atual f_t de acordo com uma regra de correção específica.

Com o objetivo de derivar essa regra de correção segue-se as ideias do algoritmo Perceptron (ROSENBLATT, 1958) usando a abordagem da descida do gradiente estocástica. Considerando o risco empírico definido na seção 2, define-se o seguinte custo:

$$J(f) := \sum_{(y_i,x_i)\in Z_m} \ell(y_i, x_i, f),$$

que deve ser minimizado em relação a f. Assim, para cada par de pontos (y_i, x_i), a seguinte regra de correção é aplicada à hipótese atual f_t

$$f_{t+1} \longleftarrow f_t - \eta \partial_f \ell(y_i, x_i, f) \tag{5.1}$$

em que $\eta > 0$ é geralmente chamada de taxa de aprendizado e ∂_f denota o gradiente da função de perda em relação a f.

Um aspecto importante dessa abordagem é que se $\ell(\cdot) \geq 0$, o que é verdadeiro para a maioria das funções de perda, a atualização acima precisa ser efetuada somente nos casos em que $\ell(y_i, x_i, f) > 0$. Caso contrário, a hipótese atual f_t já atingiu o mínimo para o exemplo (y_i, x_i) e não é necessário proceder qualquer correção, i.e., $f_{t+1} = f_t$. Nesse sentido, funções de perda que são baseadas na conceito de tubo são bem adequadas para esse esquema, uma vez que o exemplo somente afetará a hipótese atual caso encontre-se fora do tubo.

5.1 PERCEPTRON DE ε-RAIO FIXO

Para construir o algoritmo, aplica-se as ideias da seção anterior à função de perda ℓ_ε, restringindo a classe de funções $\mathcal{C}$ a funções lineares $f_{(w,b)}$. Assim, a condição $\ell_\varepsilon(\cdot) > 0$

para atualizar a hipótese $f_{(w_t,b_t)}$ após o exemplo (y_i, x_i) é:

$$|y_i - \langle w_t, x_i \rangle - b_t| > \varepsilon. \tag{5.2}$$

Para a regra de correção, o gradiente na equação (5.1) é tomado em relação aos parâmetros (w, b) que compõem a função $f_{(w,b)}$. Por isso:

$$\begin{aligned} w_{t+1} &\longleftarrow w_t + \eta \ \text{sign}(y_i - \langle w_t, x_i \rangle - b_t) x_i \\ b_{t+1} &\longleftarrow b_t + \eta \ \text{sign}(y_i - \langle w_t, x_i \rangle - b_t), \end{aligned} \tag{5.3}$$

em que $\text{sign}(x) := x/|x|$, para $x \in \mathbb{R} \setminus \{0\}$. Esse algoritmo é chamado de *Perceptron de ε-Raio Fixo* (εPRF). Um algoritmo similar foi proposto por (KIVINEN *et al.*, 2004), usando uma função de perda semelhante. Os algoritmos são equivalentes quando o parâmetro ν, usado por (KIVINEN *et al.*, 2004), é definido como zero. O algoritmo εPRF é apresentado em detalhes no Algoritmo 1.

Algoritmo 1: εPRF em variáveis primais.

input : Z_m, w_{init}, b_{init}, η, ε, T
output: (w, b)

```
1 w_0 ← w_init, b_0 ← b_init, t ← 0
2 repeat
3     for i = 1, ..., m do
4         if |y_i − ⟨w_t, x_i⟩ − b| > ε then
5             w_{t+1} ← w_t + η (sign(y_i − ⟨w_t, x_i⟩ − b_t) x_i)
6             b_{t+1} ← b_t + η (sign(y_i − ⟨w_t, x_i⟩ − b))
7         t ← t + 1
  until nenhum erro foi cometido ou t > T
8 return
```

5.1.1 PROVA DE CONVERGÊNCIA

A prova de convergência desenvolvida aqui segue os passos do teorema da convergência do algoritmo perceptron apresentado por (NOVIKOFF, 1963). Ela garante que o εPRF convergirá em um número finito de iterações. Para o teorema a seguir define-se: $R := \max_{i \in \{1,\ldots,m\}} ||x_i||$, $M := \max_{i \in \{1,\ldots,m\}} \{s_{t,i} y_i\}$ e $m := \min_{i \in \{1,\ldots,m\}} \{s_{t,i} y_i\}$, em que $s_{t,i} := \text{sign}(y_i - \langle w_t, x_i \rangle - b_t)$.

Teorema 5.1.1. *(Convergência εPRF): Dado um conjunto de treinamento Z_m e considerando uma solução (w^*, b^*), com um tubo de tamanho ε^* contendo os dados, o número de correções feitas pelo εPRF é limitada por*

$$t < \frac{2\eta^{-1}(M-\varepsilon) + R^2}{(m-\varepsilon^*)^2},$$

Demonstração. Esta prova é construída de maneira similar ao teorema da convergência do perceptron. Seja w_t o vetor normal ao hiperplano e b_t o bias após a t-ésima correção. Suponha que essa correção ocorre para o i-ésimo exemplo. Lembre que a condição para um erro no i-ésimo exemplo é dado pela equação (5.2), que é equivalente a:

$$s_{t,i}(y_i - \langle w_t, x_i \rangle - b_t) > \varepsilon,$$

de onde tem-se:

$$\langle w_t, x_i \rangle < s_{t,i}(y_i - b_t) - \varepsilon. \tag{5.4}$$

Além disso, note que para uma solução ótima (w^*, b^*) e ε^* tem-se:

$$\begin{aligned} |y_i - \langle w^*, x_i \rangle - b^*| \leq \varepsilon \Rightarrow \\ -\varepsilon^* \leq y_i - \langle w^*, x_i \rangle - b^* \leq \varepsilon^* \Rightarrow \\ -\varepsilon^* - y_i + b^* \leq -\langle w^*, x_i \rangle \leq \varepsilon^* - y_i + b^* \Rightarrow \quad (5.5) \\ \varepsilon^* + (y_i - b^*) \geq \langle w^*, x_i \rangle \geq (y_i - b^*) - \varepsilon^*, \quad (5.6) \end{aligned}$$

Usando o lado esquerdo de (5.5) e o lado direito de (5.6) leva à:

$$s_{t,i} \langle w^*, x_i \rangle \geq s_{t,i}(y_i - b^*) - \varepsilon^*, \tag{5.7}$$

para qualquer que seja o valor de $s_{t,i}$. A partir da correção dada pela equação (5.3) e usando a equação (5.4) tem-se o seguinte:

$$\begin{aligned} ||w_{t+1}|| &= ||w_t||^2 + 2\eta s_{t,i} \langle w_t, x_i \rangle + \eta^2 ||x_i||^2 \\ &< ||w_t||^2 + 2\eta s_{t,i}(y_i - b_t) - \varepsilon + \eta^2 ||x_i||^2 \\ &< \ldots < 2\eta t(M - \varepsilon) + \eta^2 t R^2. \end{aligned} \tag{5.8}$$

Ainda, a equação de correção dada por (5.2) leva a seguinte equação para o produto interno $\langle w^*, w_{t+1}\rangle$:

$$\begin{aligned}\langle w^*, w_{t+1}\rangle &= \langle w^*, w_t\rangle + \eta s_{t,i}\langle w^*, x_i\rangle \\ &\geq \langle w^*, w_t\rangle + \eta(s_{t,i}(y_i - b^*) - \varepsilon^*) \\ &\geq \ldots \geq \eta t(m - \varepsilon^*),\end{aligned} \tag{5.9}$$

em qua a expressão foi aplicada recursivamente e dado o fato que $(w_0, b_0) \equiv 0$. Agora, combinando as equações (5.8) e (5.9) e aplicando a inequação de Cauchy-Schwarz tem-se:

$$\begin{aligned}\eta t(m - \varepsilon^*) \leq \langle w^*, w_{t+1}\rangle &\leq ||w^*||||w_{t+1}|| \\ &< ||w^*||\sqrt{2\eta t(M - \varepsilon) + \eta^2 R^2} \Rightarrow \\ \eta^2 t^2 (m - \varepsilon^*)^2 &< 2\eta t(M - \varepsilon) + \eta^2 t R^2\end{aligned}$$

de onde segue que

$$t < \frac{2\eta^{-1}(M - \varepsilon) + R^2}{(m - \varepsilon^*)^2}.$$

□

5.2 PERCEPTRON DE ρ-RAIO FIXO

Para construir o algoritmo de regressão ortogonal, considera-se a função de perda ρ-insensível apresentada na Seção 2. Seguindo uma derivação análoga, a condição para atualizar a hipótese após examinar o exemplo (y_i, x_i) é:

$$\frac{|y_i - \langle w_t, x_i\rangle - b_t|}{||(1, w_t)||} > \rho.$$

A regra de correção correspondente tem a seguinte forma:

$$\begin{aligned}w_{t+1} &\longleftarrow w_t\lambda_t + \eta\left(\frac{\text{sign}(y_i - \langle w_t, x_i\rangle - b_t)x_i}{||(1, w_t)||}\right) \\ b_{t+1} &\longleftarrow b_t + \eta\left(\frac{\text{sign}(y_i - \langle w_t, x_i\rangle - b_t)}{||(1, w_t)||}\right),\end{aligned} \tag{5.10}$$

em que λ_t é dado por

$$\lambda_t := \left(1 + \eta \frac{|y_i - \langle w_t, x_i \rangle - b_t|}{||(1, w_t)||^3}\right). \tag{5.11}$$

Esse algoritmo recebe o nome de *Perceptron de ρ-Raio Fixo* (ρPRF). Ele é apresentado em detalhes no Algoritmo 2.

Algoritmo 2: ρPRF em variáveis primais.

input : Z_m, w_{init}, b_{init}, η, ρ, T
output: (w, b)

$w_0 \leftarrow w_{init}$, $b_0 \leftarrow b_{init}$, $t \leftarrow 0$
repeat
  for $i = 1, \ldots, m$ **do**
    if $|y_i - \langle w_t, x_i \rangle - b| > \rho||(1, w_t)||$ **then**
      $w_{t+1} \leftarrow w_t \lambda_t + \eta \left(\frac{\text{sign}(y_i - \langle w_t, x_i \rangle - b_t) x_i}{||(1, w_t)||}\right)$
      $b_{t+1} \leftarrow b_t + \eta \left(\frac{\text{sign}(y_i - \langle w_t, x_i \rangle - b)}{||(1, w_t)||}\right)$
    $t \leftarrow t + 1$
until *nenhum erro foi cometido* ***ou*** $t > T$
return

6 ALGORITMO DUAL

Suponha agora que o exemplos de treinamento estão em algum espaço abstrato $\mathcal{X}$. Além disso, suponha que as funções $f \in \mathcal{C}$ aceitam a seguinte representação: $f = f_{\mathcal{H}} + b$, para algum $f_{\mathcal{H}} \in \mathcal{H}$ e $b \in \mathbb{R}$, em que $\mathcal{H}$ é um *espaço de Hilbert de reprodução* (RKHS) (e.g., (SMOLA e SCHÖLKOPF, 2002)). Seja, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ e $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ o produto interno associado e o kernel, respectivamente. Então, a propriedade reprodutiva de k implica que $k(x, \cdot) \in \mathcal{H}$ e, para qualquer $f \in \mathcal{H}$, tem-se que $\langle f, k(x, \cdot) \rangle_{\mathcal{H}} = f(x)$ para todo $x \in \mathcal{X}$. Outra propriedade interessante do RKHS é que qualquer $f \in \mathcal{H}$ pode ser escrito como uma combinação linear de $k(x, \cdot)$. Esse fato é muito útil para algoritmos de aprendizado, uma vez que é possível escrever a hipótese na iteração t como:

$$f_t(x) = \sum_{i=1}^{m} \alpha_{t,i} k(x_i, x) + b_t \tag{6.1}$$

para algum $\alpha_t := (\alpha_{t,1}, \ldots, \alpha_{t,m})' \in \mathbb{R}^m$, $b_t \in \mathbb{R}$, $x \in \mathcal{X}$ e $x_i \in X_m$. Nesse sentido, pode-se definir $w_t := \sum_{i=1}^{m} \alpha_{t,i} k(x_i, \cdot)$ e interpretar a função f_t, dada na equação (6.1), na forma:

$$f_t(x) = \langle w_t, k(x, \cdot) \rangle_{\mathcal{H}} + b_t, \tag{6.2}$$

pela propriedade reprodutiva de k. Seja $|| \cdot ||_{\mathcal{H}}$ a norma induzida pelo produto interno $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, i.e. $||f||^2_{\mathcal{H}} := \langle f, f \rangle_{\mathcal{H}}$ para todo $f \in \mathcal{H}$. Então, a norma de w_t pode ser escrita como:

$$||w_t||^2_{\mathcal{H}} := \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_{t,i} \alpha_{t,j} k(x_i, x_j),$$

pela propriedade reprodutiva.

Geralmente, na prática, a construção acima da classe de funções $\mathcal{C}$ é estabelecida escolhendo-se uma função $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, que intuitivamente *mede similaridades* entre pontos em $\mathcal{X}$. Se essa função k atende às condições de Mercer (e.g., (SMOLA e SCHÖLKOPF, 2002)), é possível mostrar que existe um RKHS correspondente $\mathcal{H}$ que tem k como seu kernel associado. Quando $\mathcal{X} = \mathbb{R}^n$, uma possível escolha para k é o produto interno $\langle \cdot, \cdot \rangle$ de $\mathbb{R}^n$. Isso conduz à representação linear de f usada nas seções anteriores.

6.1 εPRF DUAL

Dada a representação acima de w_t como a combinação linear $\sum_{i=1}^{m} \alpha_{t,i} k(x_i, \cdot)$, é possível derivar a regra de correção para o algoritmo εPRF em variáveis duais α_t examinando-se a regra de correção dada pela equação (5.3). Para um erro no exemplo (y_i, x_i) a regra de correção para w_t será:

$$\sum_{j=1}^{m} \alpha_{t+1,j} k(x_j, \cdot) \quad \longleftarrow \sum_{j=1}^{m} \alpha_{t,j} k(x_j, \cdot) + \eta \ \text{sign}(y_i - f_t(x_i)) k(x_i, \cdot),$$

o que implica na seguinte regra de correção para a variável dual α_t:

$$\alpha_{t+1,i} \quad \longleftarrow \alpha_{t,i} + \eta \ \text{sign}(y_i - f_t(x_i)). \tag{6.3}$$

O algoritmo εPRF em variáveis duais é apresentado no algoritmo 3.

Algoritmo 3: εPRF em variáveis duais

input : Z_m, α_{init}, b_{init}, η, ε, T
output: α, b

$\alpha_0 \leftarrow \alpha_{init}$, $b_0 \leftarrow b_{init}$, $t \leftarrow 0$
calcular $f_0(x_j)$, for $j = 1, \ldots, m$
repeat
    for $i = 1, \ldots, m$ **do**
        if $|y_i - f_t(x_i)| > \varepsilon$ **then**
            $\alpha_{t+1,i} \leftarrow \alpha_{t,i} + \eta \left(\text{sign}(y_i - f_t(x_i))\right)$
            $b_{t+1} \leftarrow b_t + \eta \left(\text{sign}(y_i - f_t(x_i))\right)$
            atualizar $f_{t+1}(x_j)$, para $j = 1, \ldots, m$.
        $t \leftarrow t + 1$
until *nenhum erro foi cometido* ***ou*** $t > T$
return

6.2 ρPRF DUAL

A regra de correção para o ρPRF em variáveis duais é construída de maneira similar. Seguindo a atualização dada na equação (5.10), observe que w_t é escalonado por um fator λ_t, dado pela equação (5.11). Em variáveis duais isso corresponde a escalonar o vetor α_t pelo mesmo fator λ_t antes que a componente associada α_t seja corrigida. Por isso, se o i-ésimo exemplo (y_i, x_i) encontra-se fora do ρ-tubo, a atualização é feita da seguinte

maneira: primeiro α_t é escalonado por λ_t, e em seguida a correção é realizada:

$$\alpha_{t+1,i} \longleftarrow \alpha_{t,i} + \eta \left(\frac{\text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} \right),$$

em que define-se $||(1, w_t)|| := \sqrt{1 + ||w_t||^2_{\mathcal{H}}}$.

O algoritmo ρPRF em variáveis duais é apresentado no algoritmo 4.

Algoritmo 4: ρPRF em variáveis duais

input : Z_m, α_{init}, b_{init}, η, ρ, T
output: α, b

$\alpha_0 \leftarrow \alpha_{init}$, $b_0 \leftarrow b_{init}$, $t \leftarrow 0$
calcular $f_0(x_j)$, para $j = 1, \ldots, m$, e $||w_0||_{\mathcal{H}}$.
repeat
  for $i = 1, \ldots, m$ **do**
    if $|y_i - f_t(x_i)| > \rho||(1, w_t)||$ **then**
      $\alpha_{t+1} \leftarrow \lambda_t \alpha_t$
      $\alpha_{t+1,i} \leftarrow \alpha_{t,i} + \eta \left(\frac{\text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} \right)$
      $b_{t+1} \leftarrow b_t + \eta \left(\frac{\text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} \right)$
      atualizar $f_t(x_j)$ e $||w_t||_{\mathcal{H}}$.
    $t \leftarrow t + 1$
until *nenhum erro foi cometido* ***ou*** $t > T$
return

6.3 OTIMIZAÇÕES COMPUTACIONAIS

Com o objetivo de melhorar a eficiência computacional dos algoritmos, é possível atualizar os valores funcionais $f_t(x_j)$, para $j = 1, \ldots, m$, e a norma $||w_t||_{\mathcal{H}}$ a partir dos seus valores anteriores após cada atualização. Primeiro, suponha que uma atualização na iteração t foi feita para um erro no $i$ésimo exemplo (y_i, x_i). Então, examinando a equação de correção para α_{t+1} e a expressão de $f_t(\cdot)$ dada pela equação (6.2), pode-se calcular $f(x_j)$ a partir de seus valores anteriores como segue, para o ρPRF:

$$\begin{aligned} f_{t+1}(x_j) &= \lambda_t \langle w_t, k(x_j, \cdot) \rangle_{\mathcal{H}} + \eta \frac{\text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} k(x_i, x_j) + b_{t+1} \\ &= \lambda_t (f_t(x_j) - b_t) + \eta \frac{\text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} k(x_i, x_j) + b_{t+1}. \end{aligned} \tag{6.4}$$

Uma derivação análoga pode ser obtida para o algoritmo εPRF.

A norma $||w||^2_{\mathcal{H}}$ também pode ser computada após cada atualização. Se uma correção é feita devido a um erro no $i$ésimo exemplo, a norma $||w_{t+1}||_{\mathcal{H}}$ pode ser calculada como:

$$\begin{aligned} ||w_{t+1}||^2_{\mathcal{H}} &= \lambda_t^2 \langle w_t, w_t \rangle_{\mathcal{H}} + \frac{2\eta\lambda_t \text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} \langle w_t, k(x_i, \cdot) \rangle_{\mathcal{H}} + \frac{\eta^2 k(x_i, x_i)}{||(1, w_t)||} \\ &= \lambda_t^2 ||w_t||^2_{\mathcal{H}} + \frac{2\eta\lambda_t \text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} (f_t(x_i) - b_t) + \frac{\eta^2 k(x_i, x_i)}{||(1, w_t)||}. \end{aligned} \tag{6.5}$$

Dessa maneira, multiplicações do tipo vetor-matriz são evitadas, aumentando a eficiência computacional do método.

6.4 FLEXIBILIDADE

Dados experimentais com ruído frequentemente conduzem a problemas em que *outliers* estão presentes. Nesses casos, uma representação precisa do conjunto de treinamento pode resultar em uma hipótese com capacidade baixa de generalização. Assim, um mecanismo para ponderar o compromisso entre uma representação acurada dos dados e a capacidade de generalização da hipótese é crucial. No contexto das funções de perda com tubo, uma abordagem comum é introduzir as *margens flexíveis*, permitindo que os pontos mais extremos violem os limites do tubo.

Uma primeira formulação para introduzir margens flexíveis foi proposta para um problema de programação linear por (BENNETT e MANGASARIAN, 1992). Alguns anos depois (CORTES e VAPNIK, 1995) adaptaram esse conceito de margens flexíveis para as máquinas de vetores suporte. Nessa abordagem variáveis de folga são introduzidas às restrições do problema para permitir violação da margem. Essas variáveis de folga são, então, penalizadas na função de custo e um parâmetro C é introduzido como medida de compensação entre a quantidade de violações da margem e uma representação precisa dos dados de treinamento.

Outra abordagem para margens flexíveis consiste em somar uma constante $\lambda_{diag} > 0$ à diagonal da matriz kernel (SMOLA e SCHÖLKOPF, 2002):

$$\tilde{K} := K + \lambda_{diag} I,$$

onde a matriz kernel é definida como $K \in \mathbb{R}^{m \times m}$ com componentes $K_{ij} := k(x_i, x_j)$, para $x_i, x_j \in X_m$. É possível mostrar que esta abordagem é equivalente à introdução das variáveis de folga na formulação SVM, quando elas são penalizadas ao quadrado (SMOLA

e SCHÖLKOPF, 2002). De fato, é possivel estabelecer uma relação direta entre essa constante λ_{diag} e o parâmetro C, que é $\lambda_{diag} = 1/2C$ (veja por exemplo (CAMPBELL, 2002)). É importante ainda mencionar que tal modificação da matriz kernel é usada somente no treinamento do algoritmo e não deve ser usada no teste.

Para considerar flexibilidade da margem no método proposto, segue-se uma ideia similar somando uma constante λ_{diag} à diagonal da matriz kernel. É interessante analisar o efeito dessa constante λ_{diag} no algoritmo εPRF. Para a matriz kernel modificada, a condição para um ponto (y_i, x_i) estar localizado no interior do tubo, na iteração t, pode ser escrita como

$$-\varepsilon \leq y_i - \sum_{j=1}^{m} \alpha_{t,j} \tilde{K}_{ij} - b_t \leq \varepsilon,$$

ou de maneira equivalente, usando a definição de $\tilde{K}$:

$$-\varepsilon + \alpha_{t,i}\lambda_{diag} \leq y_i - \sum_{j=1}^{m} \alpha_{t,j} K_{ij} - b_t \leq \varepsilon + \alpha_{t,i}\lambda_{diag}.$$

Observe que $\alpha_{t,i}$ geralmente apresentará o mesmo sinal que $y_i - f_t(x_i)$, pela regra de correção dada na equação (6.3). O único caso em que isso poderia ser falso é quando (y_i, x_i) troca de lado em relação ao hiperplano após algumas atualizações. Isso implica que a flexibilidade da margem é obtida adicionando uma folga dada por $\xi_i := \alpha_{t,i}\lambda_{diag}$, às restrições do problema.

Com uma análise similar, obtém-se uma relaxação análoga das restrições para o algoritmo ρPRF:

$$-\rho||1, w_t|| + \alpha_{t,i}\lambda_{diag} \leq y_i - \sum_{j=1}^{m} \alpha_{t,j} K_{ij} - b_t \leq \rho||1, w_t|| + \alpha_{t,i}\lambda_{diag}.$$

É importante mencionar que, para λ_{diag} ter um efeito similar ao obtido no algoritmo εPRF, a matriz kernel original K deve ser usada para calcular a norma $||1, w_t||$, **não** $\tilde{K}$.

6.5 REGULARIZAÇÃO

Como destacado em (MARKOVSKY e HUFFEL, 2007), a regressão ortogonal é intrinsecamente um problema que promove desregularização das soluções candidatas. No algoritmo ortogonal proposto neste trabalho, esse efeito é observado pelo fato de que o

parâmetro de escalonamento λ_t, usado na regra de correção, será sempre estritamente maior do que um. Assim, a norma $||w_t||$ tende a crescer muito à medida que o algoritmo itera. Isso é natural, uma vez que aumentando o valor da norma, o algoritmo força os pontos de treinamento a pertencerem ao interior do tubo.

Quando a função kernel tem uma capacidade limitada de representação, esse efeito de desregularização não produz resultados indesejados, uma vez que o crescimento da norma é limitado pela capacidade do kernel na representação dos dados. Entretanto, esse efeito pode se tornar um problema nos casos em que a função kernel possui ilimitada ou ampla capacidade de representação, como no caso do kernel Gaussiano. Nesses casos, a solução final tende ao superajuste dos dados de treinamento.

Uma possível solução para controlar o crescimento da norma é minimizar o risco empírico regularizado (e.g., (KIVINEN *et al.*, 2004; HERBRICH, 2002)), dado por:

$$R_{\text{reg}}[f, Z_m] := R_{\text{emp}}[f, Z_m] + \beta\mathcal{O}(f),$$

em que $\mathcal{O}$ é chamado de regularizador, o qual penaliza a complexidade da solução f, e $\beta > 0$ é o parâmetro de regularização. A escolha mais comum para o regularizador, que é adotada neste trabalho, é $\mathcal{O}(f) := \frac{1}{2}||w||^2_{\mathcal{H}}$. Assim, a regularização penaliza o crescimento da norma e, por isso, essa estratégia deve ser usada com cautela, no sentido de não interferir no aspecto natural do problema de regressão ortogonal.

Usando o risco empírico regularizado, a equação de correção para um determinado ponto (y_i, x_i), que viola o tubo, é a mesma equação de correção anterior, dada pela equação (6.4), exceto pela pré-multiplicação do parâmetro λ_t, que é então dado por:

$$\lambda_t := 1 + \eta\left(\frac{|y_i - \langle w_t, x_i\rangle - b_t|}{||(1, w_t)||^3} - \beta\right).$$

Assim, o parâmetro β compensa o primeiro termo entre parenteses, contribuindo para o controle do parâmetro de escalonamento λ_t.

Além disso, a correção regularizada é também aplicada aos pontos (y_i, x_i) que são examinados, mas *não* violam o tubo (a mesma abordagem é feita em (KIVINEN *et al.*, 2004)). A regra de correção para esses pontos consiste no escalonamento dos valores α_t pelo fator: $\tilde{\lambda}_t := 1 - \eta\beta$. Observe que após essa correção pode-se atualizar $f_{t+1}(x_j)$ e $||w_{t+1}||$ seguindo uma derivação análoga a que leva às equações (6.4) e (6.5).

7 ESTRATÉGIA INCREMENTAL

Nesta seção apresenta-se uma estratégia incremental baseada em um algoritmo similar introduzido por (LEITE e NETO, 2008). Essa estratégia pode ser utilizada para obter soluções esparsas e também encontrar uma aproximação para o tubo mínimo contendo os dados. Para tanto, nesta seção restringe-se a discussão ao algoritmo ρPRF, embora os mesmos argumentos possam ser estendidos diretamente para o εPRF.

Dado um conjunto de treinamento Z_m e uma constante fixa ρ, o algoritmo ρPRF é capaz de encontrar um ponto (w, b) dentro do espaço de versões $\Omega(Z_m, \rho)$. Suponha que seja possível construir um tubo de raio $\tilde{\rho}$ tal que $\tilde{\rho} < \rho$ a partir de uma solução $(w, b) \in \Omega(Z_m, \rho)$ de tal maneira que o novo espaço de versões $\Omega(Z_m, \tilde{\rho})$ seja diferente de vazio. Então, o algoritmo ρPRF pode ser usado para encontrar uma sequência de raios estritamente decrescentes $\rho_0, \rho_1, \ldots, \rho_n$ tal que os espaços de versões correspondentes sejam não vazios.

Uma aplicação para tal estratégia de construir essa sequência de tubos de raio decrescentes é a identificação de *vetores suporte* ou seja, os pontos que se encontram mais distante dentre as amostras do conjunto de treinamento. Suponha, por exemplo, que o raio ρ_f é desejado para um dado problema. Então, pode-se proceder da seguinte maneira: primeiro escolhe-se um valor alto ρ_0 e progressivamente esse valor de raio é reduzido até um raio final ρ_n tal que $\rho_n \leq \rho_f$. Dessa maneira, à medida que o raio decresce, somente os pontos de treinamento que estão mais na fronteira afetarão a construção da hipótese, contribuindo para a esparsidade da solução.

Nos casos em que não deseja-se flexibilidade na margem, pode-se usar essa estratégia para obter uma aproximação do tubo mínimo que contém os dados, isto é:

$$\rho^* := \inf\{\rho : \Omega(Z_m, \rho) \neq \emptyset\}.$$

Isso pode ser feito de maneira iterativa, produzindo novos valores de raio ρ_n até que na iteração final N obtém-se $\rho_N \approx \rho^*$.

Para obter o valor do novo raio ρ_{n+1} a partir do anterior ρ_n, suponha que o ρPRF encontre uma solução em $\Omega(Z_m, \rho_n)$, chamada (w_n, b_n). Define-se então os correspondentes

raios positivo e negativo como:

$$\rho_n^+ = \max_i \left\{ \frac{y_i - \langle w_n, x_i \rangle - b_n}{||(1, w_n)||} \right\} \quad \rho_n^- = \max_i \left\{ \frac{\langle w_n, x_i \rangle + b_n - y_i}{||(1, w_n)||} \right\}. \tag{7.1}$$

Uma característica desejável para a solução final ρ^* é ter os valores de raios positivo e negativo balanceados. Dessa maneira, pode-se atualizar o valor do raio definindo:

$$\rho_{n+1} = \frac{(\rho_n^+ + \rho_n^-)}{2}. \tag{7.2}$$

Observe que esse novo raio sempre leva à uma solução factível, uma vez que

$$-\rho_n^- \leq \frac{y_i - \langle w_n, x_i \rangle - b_n}{||(1, w_n)||} \leq \rho_n^+ \quad \forall i \tag{7.3}$$

e somando $(\rho_n^- - \rho_n^+)/2$ na inequação, uma nova solução é obtida mudando somente o valor do bias.

Entretanto, em alguns casos é possível ter $\rho_n^+ \approx \rho_n^-$ e assim o novo raio não será muito diferente do valor anterior. De maneira a lidar com esse fato, usa-se então a seguinte regra para atualizar o raio:

$$\rho_{n+1} = \min \left\{ \frac{(\rho_n^+ + \rho_n^-)}{2}, (1 - \delta/2)\rho_n \right\},$$

em que um novo parâmetro δ é introduzido. Verifica-se que adotando essa regra o processo pode terminar com um espaço de versões vazio e a convergência para o ρPRF não será atingida. Por isso, estipula-se um número máximo de iterações T para o ρPRF convergir. Caso a convergência não seja alcançada em T iterações o algoritmo retorna a solução do último problema resolvido. Para tanto, o valor de δ deve ser cuidadosamente escolhido de maneira a não interferir no processo incremental.

No caso em que o raio desejado ρ_f é fornecido e deseja-se apenas obter uma solução mais esparsa, a escolha de δ deve ser tal que $\delta \leq 2(1 - \rho_f/\rho_n)$. Se uma aproximação para o tubo mínimo contendo os dados é o objetivo, então esse parâmetro deve ser escolhido de acordo com a qualidade esperada na aproximação. Isto é, se uma α-aproximação do tubo mínimo (i.e. o raio final é menor que $(1 + \alpha)\rho^*$, $\alpha \in (0, 1)$) é desejada, então δ deve ser escolhido como o valor de α. Para observar isso, suponha que tem-se a solução $(w_n, b_n) \in \Omega(Z_m, \rho_n)$, para algum $n \geq 1$, e um novo raio é construído $\rho_{n+1} = (1 - \alpha/2)\rho_n$. Suponha

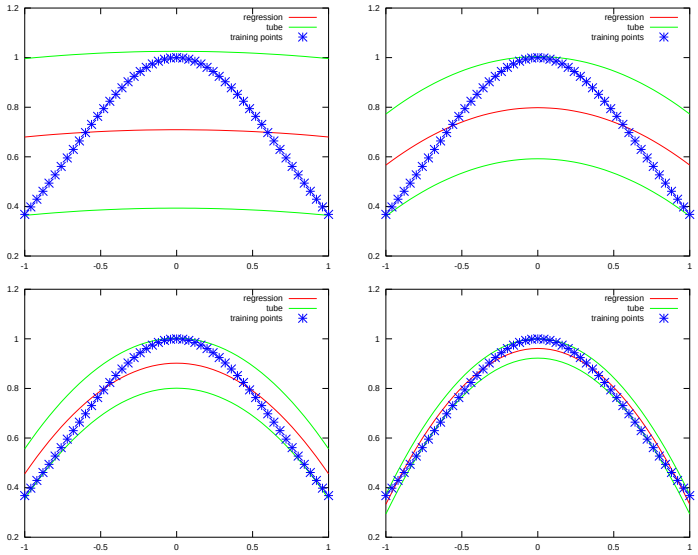


Figura 7.1: Processo do algoritmo de estratégia incremental.

que esse novo raio é tal que $\rho_{n+1} < \rho^*$. Então, o ρPRF não alcançará a convergência e a última solução factível encontrada em $\Omega(Z_m, \rho_n)$ é retornada. Essa solução final tem raio ρ_n, o que satisfaz: $\rho_n = \frac{\rho_{n+1}}{(1-\alpha/2)} < \frac{\rho^*}{(1-\alpha/2)} < (1+\alpha)\rho^*$.

Finalmente, é importante mencionar que cada solução final w do ρPRF é usada como solução inicial para o próximo problema. Essa configuração permite que o ρPRF precise fazer um pequeno número de correções para satisfazer o novo raio. Além disso, para o primeiro ρPRF, define-se o valor inicial do bias como $b_0 = \frac{1}{m}\sum_{i=1}^{n} y_i$ de maneira a auxiliar na obtenção de melhores soluções esparsas. O Algoritmo 5 apresenta a estratégia incremental usada para obter o tubo mínimo contendo os dados. O processo da estratégia incremental é ilustrado na Figura 7.1

Algoritmo 5: Algoritmo de Estratégia Incremental (AES)

input : z_m, η, δ, ρ_0, T
output: última solução factível (w_n, b_n) e o ρ_n associado

1 $w_0 \leftarrow 0$, $b_0 \leftarrow \frac{1}{m}\sum_{i=1}^{n} y_i$
2 **repeat**
3 $\quad (w_{n+1}, b_{n+1}) \leftarrow \rho\mathbf{PRF}(z_m, w_n, b_n, \eta, \rho_n, T)$
4 $\quad \rho_{n+1} = \min\left\{\frac{(\rho_n^+ + \rho_n^-)}{2}, (1-\delta/2)\rho_n\right\}$
until *a convergência do ρPRF em T iterações não foi atingida*
5 **return**

7.1 ORDENANDO OS DADOS

Observando a estratégia incremental apresentada na seção anterior, e considerando que os pontos que estão mais na fronteira dos dados são gradualmente descobertos, à medida que o raio ρ_n é reduzido, pode-se tomar vantagem desse processo e considerar inicialmente apenas os pontos mais distantes durante o loop principal do ρPRF. Embora o funcionamento descrito aqui considere apenas o algoritmo ρPRF, a extensão para o εPRF é direta.

Isso é feito através de um algoritmo simples de ordenação. Este algoritmo ordena os dados, enquanto o ρPRF itera, de acordo com a frequência que determinado ponto promove uma correção. Dessa maneira, o ρPRF pode considerar um conjunto reduzido com apenas s pontos, antes de considerar o restante dos dados.

Essa variável s é calculada iterativamente à medida que o algoritmo executa. Esse valor é iniciado como $s = 0$ e é incrementado progressivamente ao passo que o algoritmo encontra os pontos que promovem uma atualização na solução. Esse parâmetro s é passado através das diversas chamadas do ρPRF realizadas pelo algoritmo de estratégia incremental.

Além disso, um vetor de índices idx é definido para controlar a ordenação dos dados. Este vetor é inicializado com $idx \leftarrow \{1, \ldots, m\}$ e também é passado através das chamadas consecutivas do ρPRF pelo AES. O algoritmo de ordenação é apresentado no algoritmo 6.

Algoritmo 6: ρPRF usando um algoritmo simples de ordenação.

input : Z_m, w_{init}, b_{init}, η, ρ, T, s, idx
output: (w, b)

```
1  w_0 ← w_init, b_0 ← b_init, t ← 0
2  repeat
3      e ← 0
4      for k = 1, ..., m do
5          i ← idx(k)
6          if ponto (y_i, x_i) é um erro para (w_t, b_t) then
7              (w_{t+1}, b_{t+1}) ← atualiza(w_t, b_t)
8              e ← e + 1
9              if k > s then
                   s ← s + 1, j ← s
10             else
                   j ← e
11             troca(idx,j,k)
12         else if t > 1 e e > 1 e k > s then
               break
13         t ← t + 1
   until e = 0 ou t > T
14 return
```

8 EXPERIMENTOS

Este capítulo foi construído com o objetivo de cobrir as diversas características técnicas do método proposto. Nesse sentido, o capítulo está dividido em seis partes como segue: na seção 8, um aspecto interessante dos modelos de regressão ortogonal é verificado para o método proposto. A seção 8 apresenta os resultados relacionados a aplicação dos algoritmos εPRF e ρPRF combinados com a estratégia incremental para obter soluções esparsas. Também, apresenta-se comparações com o SVM-light (JOACHIMS, 1999). Além disso, discute-se a qualidade da solução sobre variação da intensidade de ruído nas variáveis. A seção 8 mostra resultados relacionados ao uso da estratégia incremental para obter uma aproximação para o tubo mínimo contendo os dados. A seção 8 traz os resultados relativos ao uso da regularização no método ρPRF. Na seção 8 compara-se o tempo de execução entre o algoritmo εPRF e o SVM-light. Finalmente a seção 8 apresenta os resultados obtidos em bases de dados de *benchmark* usadas na literatura.

8.1 TRATANDO VARIÁVEIS SIMETRICAMENTE

Uma característica interessante dos modelos de regressão ortogonal é que eles tratam as variáveis do problema de maneira simétrica (AMMANN e NESS, 1988). Esse procedimento pode ser muito útil quando o problema de fato não possui variável *independente* e *dependente*, e portanto elas devem ser tratadas igualmente.

Este primeiro exemplo mostra que o método proposto de regressão ortogonal apresenta esse comportamento. A Figura 8.1 retrata três regressões. A linha sólida corresponde a regressão ortogonal usando o algoritmo ρPRF. A linha tracejada é a regressão clássica usando o εPRF para estimar Y (como variável dependente) a partir de X (como variável independente). A linha traço-ponto descreve a regressão clássica usando o εPRF, em que as váriaveis dependente e independente foram invertidas, ou seja, estimando X a partir de Y. Além disso, a Figura 8.1 apresenta as linhas das respectivas distâncias que são medidas por cada regressão para um ponto qualquer. É importante mencionar que caso a regressão ortogonal, usando o ρPRF, seja feita com as variáveis invertidas, a mesma linha sólida é obtida.

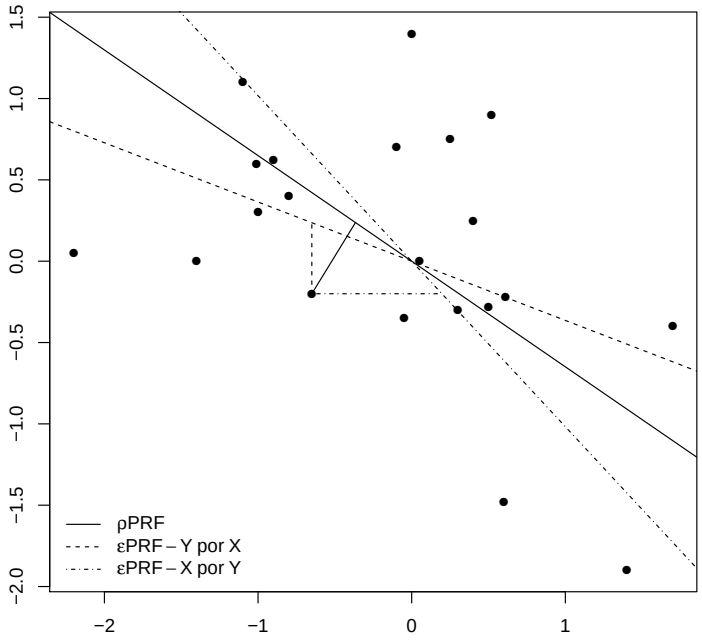


Figura 8.1: Regressão ortogonal (ρPRF) e regressão clássica (εPRF).

8.2 ESPARSIDADE

Neste grupo de experimentos os algoritmos ρPRF e εPRF foram combinados com o algoritmo de estratégia incremental (AES) e aplicados a diferentes bases de dados geradas artificialmente. Como discutido na seção 7 o AES pode ser muito útil na obtenção de soluções esparsas, uma vez que o processo encontra os pontos que contribuem para a solução de forma gradual. No sentido de observar essa característica, aplicou-se também os algoritmos sem a combinação com o AES. Os resultados obtidos executando-se o SVM-light também são apresentados para comparações.

Os conjuntos de treinamento foram gerados a partir de determinada função matemática e poluídos com diferentes intensidades de ruído em ambas as variáveis. Isso é feito para comparar a regressão ortogonal (ρPRF) e a regressão clássica (εPRF e SVM). Para os conjuntos de teste gerou-se uma nova base de dados, usando a mesma função escolhida, com diferentes pontos distribuídos sobre a mesma faixa sem introdução de ruído. O conjunto de teste possui o dobro do número de pontos em relação ao conjunto de treinamento.

Base	m	Função	Intervalo	Ruído Gaussiano
Linear1	51	$y = 2x + 0.1$	$x \in [-5, 5]$	$\sigma_x = 0.2$ e $\sigma_y = 0.2$
Linear2	51	$y = 2x + 0.1$	$x \in [-5, 5]$	$\sigma_x = 0.04$ e $\sigma_y = 0.4$
Exp1	51	$y = e^{-x^2}$	$x \in [-1, 1]$	$\sigma_x = 0.1$ e $\sigma_y = 0.1$
Poly3	61	$y = x^3$	$x \in [-3, 3]$	$\sigma_x = 0.2$ e $\sigma_y = 0.2$

Tabela 8.1: Informações sobre as bases de dados.

Os conjuntos de treinamento são descritos na tabela 8.1 para cada função escolhida.

Para comparar as soluções apresenta-se os seguintes dados obtidos a partir dos experimentos: número de vetores suporte em relação ao total de exemplos no conjunto de treinamento (vs/m), mostrado para avaliar a esparsidade da solução; raio do tubo ortogonal (ρ) e vertical (ε); norma da solução (norma $= ||(1, w)||$). Para os métodos PRF apresenta-se também o número total de iterações (it) e o número total de correções (up) realizados pelos algoritmos. Para medir a qualidade do ajuste foram usadas duas medidas de erro. A primeira é a raiz quadrada do erro médio quadrático (RMSE). Esse critério toma a raiz do erro médio quadrático, em que esse erro equivale à diferença direta entre a função estimada e os valores observados. Nesse sentido, uma segunda medida de erro é proposta baseada no ajuste ortogonal, chamada de erro médio quadrático geométrico (gMSE), dada por

$$\text{gMSE} := \frac{1}{m} \sum_{i}^{m} \frac{(y_i - \langle w, x_i \rangle - b)^2}{||(1, w)||^2},$$

e também toma-se a raiz quadrada dessa medida, derivando o critério RgMSE.

Os testes apresentados nesta seção foram realizados da seguinte maneira. Primeiro, calcula-se a faixa dos targets $r := \max_{i=1,\ldots,m} y_i - \min_{i=1,\ldots,m} y_i$ no conjunto de treinamento. Então, define-se o valor de ε como $0.1r$ para os algoritmos εPRF e SVM. Também, esse valor é usado como critério de parada para o εPRF combinado com o AES (daqui para frente chamado de $\varepsilon\text{PRF}_{AES}$). Para comparar os resultados, calcula-se o respectivo ρ obtido na solução do εPRF, pela relação $\varepsilon = \rho||(1, w)||$, e esse valor é usado para o ρPRF e como critério de parada para o ρPRF combinado com o AES (daqui para frente chamado de ρPRF_{AES}). O parâmetro de capacidade C é definido como $C/m = 10$, em que m é o número de exemplos de treinamento, como sugerido em (SMOLA e SCHÖLKOPF, 2002). Para os métodos PRF sempre é usado o algoritmo de ordenação apresentado na seção 5.

Além disso, define-se a taxa de aprendizado $\eta = 0.01$. Para as bases de dados *Linear1* e *Linear2*, foi escolhido o kernel linear $k(x_i, x_j) := \langle x_i, x_j \rangle$. Para a base *Exp1*, foi usado um kernel polinomial $k(x_i, x_j) := (s \langle x_i, x_j \rangle + c)^d$ com $d = 2$, $s = 1$ e $c = 0$. Também usou-se um kernel polinomial com $d = 3$, $s = 1$ e $c = 0$ para a base *Poly3*. Em cada caso, o modelo gerado no treinamento é salvo e usado para realizar os testes. Os critérios RMSE e RgMSE são medidos nos conjuntos de treinamento e teste. Os resultados são apresentados na tabela 8.2.

	vs/m	ρ/ε/norma	it/up	Treinamento		Teste	
				RMSE	RgMSE	RMSE	RgMSE
Linear1							
ρPRF	44/51	0,626/1,344/2,149	7/65	0,48068	0,22367	0,28514	0,13268
ρPRF_{AES}	5/51	0,625/1,327/2,122	92/46	0,56054	0,26415	0,41071	0,19354
εPRF	37/51	0,626/1,952/3,120	4/50	0,88479	0,28359	0,81896	0,26249
$\varepsilon\text{PRF}_{AES}$	6/51	0,639/1,928/3,015	74/37	1,00106	0,33201	0,94416	0,31314
SVM-light	2/51	0,998/1,928/1,932	-/-	1,09079	0,56468	1,04105	0,53893
Linear2							
ρPRF	45/51	0,624/1,302/2,087	4/61	0,60639	0,29061	0,49572	0,23757
ρPRF_{AES}	4/51	0,579/1,221/2,105	86/43	0,56098	0,26645	0,43669	0,20742
εPRF	36/51	0,624/1,936/3,10	4/49	0,90116	0,29044	0,83506	0,26913
$\varepsilon\text{PRF}_{AES}$	4/51	0,584/1,828/3,131	76/38	0,88459	0,28254	0,81595	0,26061
SVM-light	2/51	0,935/1,828/1,955	-/-	0,99727	0,51003	0,93765	0,47954
Exp1							
ρPRF	40/51	0,094/0,240/2,566	1105/5789	0,11911	0,04642	0,03528	0,01375
ρPRF_{AES}	5/51	0,094/0,230/2,460	2580/3894	0,11343	0,04611	0,03103	0,01261
εPRF	41/51	0,094/0,098/1,043	26304/164166	0,11382	0,10912	0,04679	0,04486
$\varepsilon\text{PRF}_{AES}$	22/51	0,093/0,098/1,055	25727/130023	0,11272	0,10685	0,03840	0,03640
SVM-light	22/51	0,093/0,080/1,158	-/-	0,11289	0,09748	0,04033	0,03483
Poly3							
ρPRF	5/51	4,179/5,777/1,382	65/113	1,94745	1,40884	0,53883	0,38980
ρPRF_{AES}	7/51	3,950/5,770/1,461	79/196	1,89469	1,29713	0,77797	0,53260
εPRF	9/51	4,179/5,409/1,294	12062/44740	1,96528	1,51846	1,26403	0,97665
$\varepsilon\text{PRF}_{AES}$	5/51	4,214/5,370/1,274	237/574	2,09847	1,64681	1,61558	1,26785
SVM-light	2/51	4,068/5,370/1,320	-/-	2,27268	1,72150	2,00520	1,51889

Tabela 8.2: Resultados obtidos pela regressão ortogonal (ρPRF) e clássica (εPRF e SVM-light), comparando esparsidade e qualidade da solução sob diferentes intensidades de ruído.

Em primeiro lugar observa-se que o AES se mostrou efetivo em relação a obtenção de soluções esparsas. Na maioria dos casos os métodos PRF apresentam um número bem menor de vetores suporte quando combinados com o AES. A única exceção nos exemplos anteriores é para o ρPRF com a base de dados *Poly3*. Isso pode ser explicado pelo fato de que o ρPRF_{AES} um raio menor no treinamento. Contudo, observa-se que o número de

vetores suporte é apenas um pouco maior e segue próximo ao número obtido pela solução SVM.

Em segundo lugar nota-se que, quando combinados com o AES, os algoritmos PRF mostram um número maior de iterações, como esperado, uma vez que eles iniciam com um valor alto para o raio que é gradualmente reduzido. Entretanto, quando combinados com o AES os algoritmos realizam um número menor de correções, o que sugere que apenas os pontos mais importante para a solução influenciam no número de correções.

Considerando a qualidade do ajuste os métodos de regressão ortogonal apresentaram os melhores resultados. Para a base de dados *Exp1*, os resultados são bastante similares considerando o critério RMSE, contudo os algoritmos εPRF e $\varepsilon\text{PRF}_{AES}$ realizaram um grande número de iterações e correções.

8.3 TUBO MÍNIMO

Nesta seção apresenta-se os resultados de experimentos realizados combinando os algoritmos ρPRF e εPRF com o AES sem permitir a flexibilidade da margem, com o objetivo de obter uma aproximação do tubo mínimo que contém todos os dados. É importante mencionar que esse tubo mínimo não pode ser obtido usando a abordagem tradicional da regressão-SV.

Os experimentos foram realizados usando três bases de dados descritas na tabela 8.1, *Linear1*, *Exp1* e *Poly3*. Para comparar os resultados apresenta-se o número de vetores suporte em relação ao total de exemplos no conjunto de treinamento (vs/m); raio do tubo ortogonal (ρ) e vertical (ε); norma da solução (norma = $||(1, w)||$). Além disso, apresenta-se também os valores dos erros RMSE e RgMSE medidos nos conjuntos de treinamento e teste.

Os testes foram realizados da seguinte forma: para o ρPRF_{AES} e $\varepsilon\text{PRF}_{AES}$ define-se a taxa de aprendizado $\eta = 0.01$ e o número de iterações $T = 1000$. O valor de ε obtido pelo $\varepsilon\text{PRF}_{AES}$ ao final do processo é usado para executar o SVM-light. O parâmetro de controle da capacidade C foi definido como um valor muito alto para evitar a flexibilidade da margem no caso do SVM-light. As funções kernel escolhidas são as mesmas da seção anterior. Os resultados são apresentados na tabela 8.3.

Nesse experimento, como o raio do tubo está sendo encolhido até uma aproximação do tubo mínimo, espera-se que os algoritmos obtenham resultados semelhantes. Isto pode

	sv	ρ	ε	norma	Treinamento		Teste	
					RMSE	RgMSE	RMSE	RgMSE
Linear1								
ρPRF$_{AES}$	6/51	0,34818	0,78690	2,26007	0,43324	0,19169	0,06748	0,02986
εPRF$_{AES}$	10/51	0,18559	0,79373	4,27679	0,45657	0,10676	0,13262	0,03101
SVM-light	2/51	0,34834	0,79373	2,27858	0,46253	0,20299	0,14460	0,06346
Exp1								
ρPRF$_{AES}$	4/51	0,22521	0,26368	1,17082	0,11425	0,09758	0,03105	0,02652
εPRF$_{AES}$	4/51	0,24895	0,26670	1,07129	0,11544	0,10776	0,03078	0,02873
SVM-light	2/51	0,23021	0,26670	1,15852	0,11459	0,09891	0,03614	0,03119
Poly3								
ρPRF$_{AES}$	8/61	1,00253	5,35425	5,34074	2,14634	0,40188	1,88537	0,35302
εPRF$_{AES}$	6/61	3,99057	5,21344	1,30644	1,84730	1,41399	0,90730	0,69449
SVM-light	2/61	3,92966	5,21344	1,32669	2,29019	1,72625	1,96983	1,48477

Tabela 8.3: Resultados da regressão ortogonal (ρPRF) e clássica (εPRF e SVM-light) sem permitir flexibilidade na margem para obter uma aproximação para o tubo mínimo contendo os dados.

ser observado para as bases de dados *Linear1* e *Exp1*, em que as medidas de erro são bem próximas. Para a base de dados *Poly3* observa-se que o εPRF$_{AES}$ obteve o melhor resultado para a medida RMSE e o ρPRF$_{AES}$ o melhor resultado considerando o erro RgMSE.

8.4 INTRODUZINDO REGULARIZAÇÃO

Como discutido na seção 11, a regressão ortogonal é intrinsecamente um processo que promove uma desregularização nas soluções candidatas e isto pode se tornar um problema quando funções kernel com uma grande capacidade de representação são usadas. Nesta seção, apresenta-se os resultados obtidos em experimentos com o ρPRF usando o kernel Gaussiano

$$k(x_i, x_j) := exp(-\gamma||x_i - x_j||^2), \tag{8.1}$$

e introduzindo a regularização apresentada seção 11.

O conjunto de treinamento foi gerados a partir de determinada função matemática e poluído com ruído em ambas as variáveis. Como conjunto de teste gerou-se uma nova base de dados, usando a mesma função escolhida, com diferentes pontos distribuídos sobre a mesma faixa sem introdução de ruído. O conjunto de teste possui o dobro do número de pontos em relação ao conjunto de treinamento.

A base de dados utilizada, chamada de *Sinc*, foi gerada com $m = 63$ pontos a partir da função $y = \text{sinc}(x)$, em que $\text{sinc}(x) := \sin(x)/x$ e $x \in [-\pi, \pi]$. Foi adicionado um ruído

Gaussiano tanto nos pontos x, como nos targets y com desvio padrão $\sigma_x = \sigma_y = 0.2$.

Para comparar os resultados apresenta-se o número de vetores suporte em relação ao total de exemplos no conjunto de treinamento (vs/m); raio do tubo ortogonal (ρ) e vertical (ε); norma da solução (norma = $||(1, w)||$). Além disso, apresenta-se também os valores dos erros RMSE e RgMSE medidos nos conjuntos de treinamento e teste. Para o parâmetro de regularização β foram usados quatro valores diferentes: (i) $\beta = 0.001$, (ii) $\beta = 0.005$, (iii) $\beta = 0.01$ and (iv) $\beta = 0.02$. O algoritmo regularizado foi chamado de ρPRF$_{AES}$-reg.

Os experimentos desta seção foram realizados da seguinte maneira: primeiro, foi executado o ρPRF$_{AES}$ e ρPRF$_{AES}$-reg fazendo o número de iterações como $T = 1000$ e $T = 5000$. A taxa de aprendizado foi definida como $\eta = 0.01$. Para executar o SVM-light toma-se o ε obtido com ρPRF$_{AES}$-reg que gerou o melhor resultado . O parâmetro de controle de capacidade foi definido como $C/m = 10$. O parâmetro do kernel Gaussiano foi definido como $\gamma = 1$. Os erros RMSE e RgMSE foram computados no treinamento e teste. Os resultados são apresentados na tabela 8.4. Além disso, a figura 8.2 ilustra a solução para $T = 5000$.

			Treinamento		Teste	
	vs/m	$\rho/\varepsilon/norma$	RMSE	RgMSE	RMSE	RgMSE
Sinc - ($T = 1000$)						
ρPRF$_{AES}$	14/63	0,18699/0,35855/1,91750	0,22058	0,11503	0,17869	0,09319
ρPRF$_{AES}$-reg(i)	14/63	0,19485/0,35597/1,82689	0,21589	0,11817	0,16089	0,08806
ρPRF$_{AES}$-reg(ii)	12/63	0,21893/0,35785/1,63453	0,20658	0,12638	0,11808	0,07224
ρPRF$_{AES}$-reg(iii)	10/63	0,26349/0,37910/1,43873	0,20840	0,14485	0,11157	0,07755
ρPRF$_{AES}$-reg(iv)	6/63	0,40338/0,49470/1,22638	0,25835	0,21066	0,16254	0,13254
SVM-light	6/63	0,26647/0,37910/1,42265	0,21025	0,14778	0,10802	0,07593
Sinc - ($T = 5000$)						
ρPRF$_{AES}$	24/63	0,08280/0,32418/3,91510	0,21568	0,05509	0,18522	0,04731
ρPRF$_{AES}$-reg(i)	21/63	0,11361/0,32843/2,89093	0,22022	0,07618	0,18170	0,06285
ρPRF$_{AES}$-reg(ii)	17/63	0,16136/0,33366/2,06782	0,21277	0,10289	0,13467	0,06513
ρPRF$_{AES}$-reg(iii)	9/63	0,23142/0,35414/1,53029	0,21209	0,13860	0,11383	0,07439
ρPRF$_{AES}$-reg(iv)	6/63	0,40816/0,49540/1,21376	0,26224	0,21606	0,16730	0,13784
SVM-light	7/63	0,23039/0,35414/1,53709	0,21512	0,13995	0,11123	0,07236

Tabela 8.4: Resultados obtidos na execução do ρPRF$_{AES}$ e ρPRF$_{AES}$-reg com 1000 e 5000 iterações para a base de dados *Sinc*. Comparações com o SVM-light são apresentadas

Observa-se que, com mais iterações, a norma do ρPRF$_{AES}$ cresce e o erro RgMSE diminui. Entretanto, a figura 8.2 mostra que o ρPRF$_{AES}$ gera uma curva retorcida. Após a introdução da regularização, a curva é suavizada. Além disso, é interessante notar que escolhendo-se uma penalização alta, a solução tende a ser muito simples e não satisfaz como ilustrado no caso do ρPRF$_{AES}$-reg(iv) na figura 8.2.

Além disso, observe que é importante escolher o parâmetro de regularização cuidadosa-

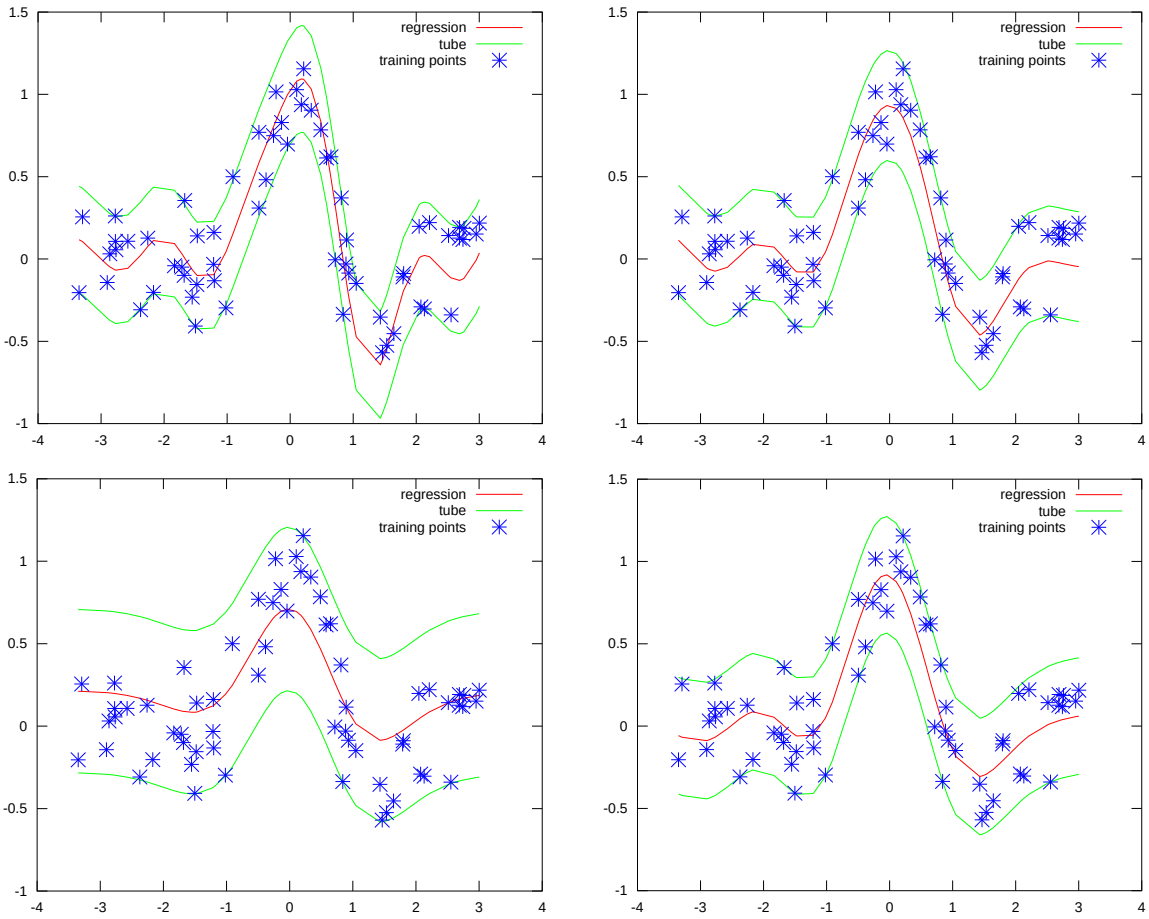


Figura 8.2: *Superior à esquerda*: ρPRF$_{AES}$. *Superior à direita*: ρPRF$_{AES}$-reg(iii). *Inferior à esquerda*: ρPRF$_{AES}$-reg(iv). *Inferior à direita*: SVM-light

mente. Na tabela 8.4 é possível observar que o ρPRF$_{AES}$-reg(i), que usa um pequeno valor de β, tem uma solução pouco penalizada e portanto o resultado é similar ao ρPRF$_{AES}$. Por outro lado, o ρPRF$_{AES}$-reg(iv) possui um alto valor de β, assim a solução é altamente penalizada e o resultado obtido não é o desejado. Nesse sentido, deve-se escolher um valor adequado para β, que produz uma boa solução. Em particular, para o problema desta seção, os melhores resultados foram obtidos com as escolhas (ii) e (iii) e esse valor pode variar de acordo com o problema.

8.5 TEMPO DE EXECUÇÃO

Nesta seção testou-se o algoritmo εPRF$_{AES}$ em grandes bases de dados geradas artificialmente. Isto é feito com o objetivo de comparar o tempo de execução do εPRF$_{AES}$ com a com o tempo demandado pela solução SVM obtida pelo algoritmo SVM-light. Além

disso, o algoritmo desenvolvido para regressão ortogonal não é considerado nessa seção no sentido de avaliar apenas o εPRF$_{AES}$, como alternativa ao SVM (ou seja, considerando apenas regressão clássica), principalmente para aplicações em larga escala ou que dependem de tempo.

Os conjuntos de treinamento foram gerados a partir de determinada função matemática e os targets foram poluídos com ruído Gaussiano. As informações sobre os conjuntos de dados gerados para esse grupo de experimentos são apresentados na tabela 8.5. A figura 8.3 ilustra a relação entre as variáveis dos conjuntos gerados para este experimento.

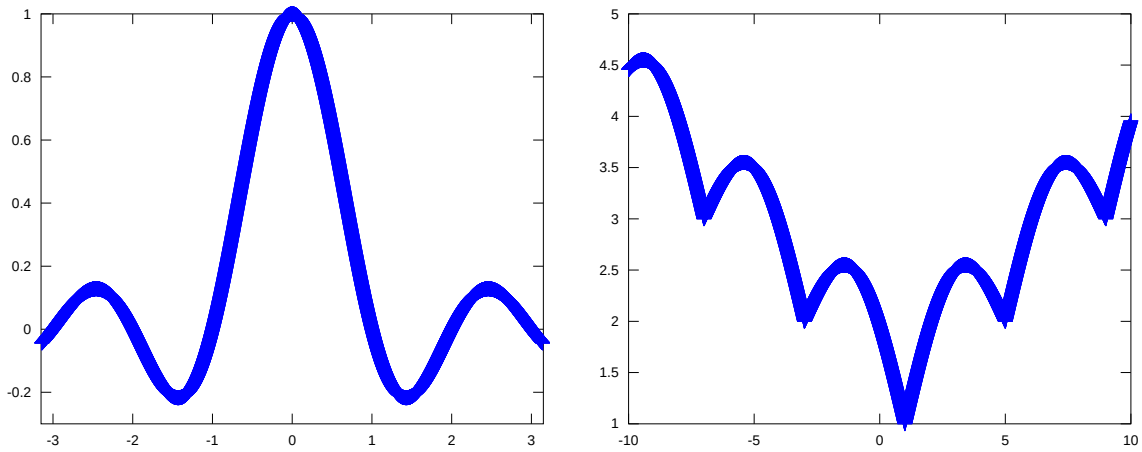
Base de dados	m	Função	Intervalo	Ruído Gaussiano
F_1	10006	$y = \text{sinc}(x)$	$x \in [-\pi, \pi]$	$\sigma_y = 0.1$
F_2	10001	$y = \left\lvert\frac{x-1}{4}\right\rvert + \left\lvert\text{sen}(\pi(1 + \frac{x-1}{4}))\right\rvert + 1$	$x \in [-10, 10]$	$\sigma_y = 0.1$
F_3	10001	$y = \text{sinc}(\sqrt{x_1^2 + x_2^2})$	$x_1, x_2 \in [-10, 10]$	$\sigma_y = 0.1$

Tabela 8.5: Informações sobre as bases de dados.

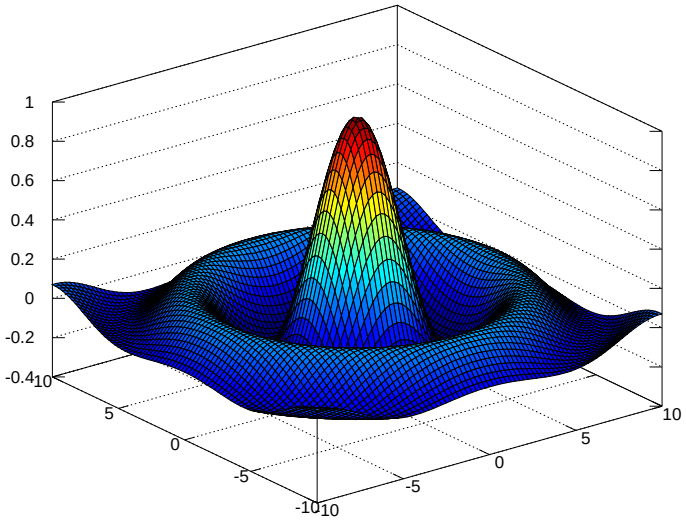
Os experimentos realizados nessa seção foram executados da seguinte maneira: Para o εPRF$_{AES}$ definiu-se como critério de parada o raio do tubo $\varepsilon_{F_1} = 0.1$ e taxa de aprendizado $\eta_{F_1} = 0.01$, para a base de dados F_1. Para o conjunto F_2 definiu-se como critério de parada o raio $\varepsilon_{F_2} = 0.2$ e taxa de aprendizado $\eta_{F_2} = 0.03$. Para a base F_3 foi definido $\varepsilon_{F_3} = 0.1$ e $\eta_{F_3} = 0.02$. Em todos os casos o valor obtido como raio final foi usado para executar o SVM-light, no sentido de avaliar o tempo gasto para obter a solução com mesmo raio. O parâmetro de capacidade foi definido como $C = 10$ para todos os testes. O kernel utilizado foi o Gaussiano, como na equação (8.1), com $\gamma = 1.0$.

Para comparar os resultados são apresentados os seguintes dados: número total de vetores suportes (vs); o tempo de execução dos métodos (rt); raio do tubo (ε); norma da solução (norma = $||w||$); para o εPRF$_{AES}$ apresenta-se também o número total de iterações (it) e número total de correções (up). Os resultados são apresentados na tabela 8.6.

Observe que para o εPRF$_{AES}$ são apresentadas soluções incrementais no processo de execução do algoritmo. Isso destaca um importante aspecto do método. A qualquer momento a execução pode ser interrompida e o εPRF$_{AES}$ retorna uma solução viável obtida até o instante da parada. Essa característica torna o método interessante para aplicações em que a solução completa do problema não se faz necessária, antes, deseja-se obter uma aproximação da solução de maneira mais rápida.



(a) F_1 (b) F_2



(c) F_3

Figura 8.3: Relação entre os pontos de treinamento e targets para os conjuntos de dados gerados.

Como esperado, em todos os casos observa-se que a obtenção da solução final usando o εPRF$_{AES}$ é mais rápida do que a solução SVM. Além disso, é possível obter uma boa solução rapidamente com o εPRF$_{AES}$. Contudo, à medida que a solução se aproxima do raio fixado como critério de parada o tempo de processamento aumenta, uma vez que a cada redução no valor de ε o espaço de versões se torna menor, e a solução mais difícil de ser obtida.

	vs	it/up	ε	norma	rt
F_1					
SVM-light	3322	-/-	0,09949	1,45873	791,48
εPRF$_{AES}$	27	103/138	0,49859	0,71759	8,58
εPRF$_{AES}$	33	163/222	0,39910	0,93835	8,66
εPRF$_{AES}$	66	457/1510	0,29898	1,37356	9,63
εPRF$_{AES}$	600	1470/40446	0,19904	5,19886	37,24
εPRF$_{AES}$	1798	2103/138435	0,14974	10,29991	106,36
εPRF$_{AES}$	3848	2914/381342	0,09949	18,64224	277,73
F_2					
SVM-light	571	-/-	0,19846	4,54684	32,43
εPRF$_{AES}$	69	67/142	0,98938	2,12675	11,26
εPRF$_{AES}$	75	118/205	0,68016	2,54115	11,32
εPRF$_{AES}$	85	167/265	0,49988	2,93399	11,39
εPRF$_{AES}$	173	412/1498	0,29944	3,66530	12,37
εPRF$_{AES}$	391	678/6132	0,24400	4,80350	15,86
εPRF$_{AES}$	1034	906/23449	0,19846	7,81095	28,77
F_3					
SVM-light	3555	-/-	0,09543	1,73510	1836,48
εPRF$_{AES}$	27	75/88	0,49038	0,79215	12,90
εPRF$_{AES}$	33	107/127	0,39383	1,03416	12,93
εPRF$_{AES}$	66	264/643	0,29984	1,38644	13,37
εPRF$_{AES}$	823	1286/24495	0,19981	5,57679	31,92
εPRF$_{AES}$	1992	1884/82775	0,14817	11,28930	75,07
εPRF$_{AES}$	4497	2462/238715	0,09543	21,20861	190,40

Tabela 8.6: Resultados comparando tempo de execução em grandes bases de dados entre o εPRF$_{AES}$ e o SVM-light.

8.6 *BENCHMARK*

Nesta seção testou-se o método ρPRF em três diferentes bases de dados de *benchmark*: BostonHousing, Quake e DEE. A primeira base de dados foi obtida no repositório da UCI Machine Learning (FRANK e ASUNCION, 2010), e as outras no repositório KEEL (ALCALá-FDEZ *et al.*, 2011). A tabela 8.7 apresenta um pequeno sumário das três bases.

Base de dados	#exemplos	#atributos
BostonHousing	506	13
Quake	2178	3
DEE	365	6

Tabela 8.7: Informações sobre as bases de dados

Em todos os experimentos desta seção foi usado o kernel Gaussiano (8.1). Os dados

de entrada $x_i = (x_{i1}, \ldots, x_{in}) \in X_m$ foram normalizados linearmente

$$x_i = \left(\frac{x_{i1} - \min\limits_{1 \leq k \leq m} x_{k1}}{\max\limits_{1 \leq k \leq m} x_{k1} - \min\limits_{1 \leq k \leq m} x_{k1}} \right),$$

porém nenhuma normalização foi feita nos targets.

Para as três bases de dados foram selecionados aleatoriamente 60% do total de exemplos como conjunto de treinamento, 20% como conjunto de validação e 20% como conjunto de teste.

O experimento foi feito da seguinte maneira: executou-se o ρPRF no conjunto de treinamento de maneira a obter os melhores valores para os parâmetros γ (kernel Gaussiano) e C (capacidade) considerando o erro no conjunto de validação a partir de todas as combinações possíveis de $\gamma = \{0.05, 0.1, 0.5, 1, 2, 4, 8, 10\}$ e $C = \{0.5, 1, 10, 20, 40, 60, 100, 1000\}$. O raio do tubo foi fixado como $\rho = 0.5$ para as bases BostonHousing e DEE, e $\rho = 0.1$ para Quake. Com o objetivo de selecionar o modelo mais adequado não usou-se a estratégia incremental, para que o valor obtido para o raio do tubo seja sempre o mesmo em todas as execuções do algoritmo. A taxa de aprendizado foi definida como $\eta = 0.02$. O modelo selecionado foi aplicado no conjunto de teste e os critérios de erro RMSE e RgMSE medidos.

Para comparar a solução, os mesmos parâmetros selecionados e o ε correspondente foram usados para executar o SVM-light e os erros RMSE e RgMSE também foram medidos no conjunto de teste. A tabela 8.8 apresenta os resultados dos seguintes dados obtidos a partir dos experimentos: percentual de vetores suportes em relação ao total de exemplos de treinamento ($\%vs$); a norma (norma = ||(1,w)||); critérios de erro RMSE e RgMSE medidos no conjunto de teste.

Em primeiro lugar nota-se que o ρPRF apresentou um elevado número de vetores suportes. Isso se justifica pelo fato de não ter sido usada a estratégia incremental. Em relação à qualidade da solução observa-se que o ρPRF obteve sempre o melhor resultado para o critério RgMSE e de maneira geral uma boa solução considerando o RMSE, seguindo próximo ao valor obtido pelo SVM-light para as bases BostonHousing e Quake.

	$\%vs$	norma	RMSE	RgMSE
BostonHousing				
ρPRF	41,58	3,45663	1,180746	0,341589
SVM-light	5,94	1,66134	1,023353	0,615980
Quake				
ρPRF	58,19	4,30836	0,365258	0,084779
SVM-light	3,37	1,00289	0,310748	0,309850
DEE				
ρPRF	74,88	2,98918	1,521271	0,508926
SVM-light	5,48	1,53335	0,865388	0,564376

Tabela 8.8: Informações sobre as bases de dados

9 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado um algoritmo online para regressão clássica, similar a ideias anteriormente apresentadas na literatura, usando a função de perda ε-insensível. Para esse algoritmo foi apresentada uma nova prova de convergência que garante um número finito de correções.

Além disso, foi introduzida uma nova formulação para regressão ortogonal baseada numa abordagem de treinamento online usando o método da descida do gradiente estocástica. O método proposto usa uma função de perda baseada na função ε-insensível, que recebeu o nome de ρ-insensível, o que possibilita a aplicação de vetores suporte. Quando formulado em variáveis duais o método permite a introdução de kernels, através do "kernel trick", e flexibilidade na margem. O algoritmo é inteiramente baseado no perceptron, o que o torna simples de entender e fácil de implementar. Até onde se sabe, este é o primeiro algoritmo online para regressão ortogonal com kernels.

Ainda, apresentou-se um algoritmo de estratégia incremental, que pode ser combinado com os algoritmos anteriores com o objetivo de obter soluções esparsas e também uma aproximação para o tubo mínimo contendo os dados.

Os resultados experimentais destacam as características dos métodos propostos. O uso da estratégia incremental realmente se mostrou válido na obtenção de soluções esparsas e também na obtenção de uma aproximação do tubo mínimo. Além disso, pode-se observar que o método de regressão ortogonal (ρPRF) obteve bons resultados em relação a regressão clássica (εPRF e SVM-light) quando o ruído foi introduzido em ambas as variáveis.

Vale destacar que a literatura ainda carece de estudos relativos a métodos online para regressão ortogonal. A abordagem apresentada nesse trabalho abre caminho para o desenvolvimento de novos métodos para regressão ortogonal e aplicações.

9.1 TRABALHOS FUTUROS

Como trabalhos futuros pretende-se introduzir novas formulações para regressão ortogonal com diferentes normas, principalmente as normas L_1 e L_∞. Além disso, uma vez que o modelo de regressão ortogonal trata simetricamente as variáveis, o uso de normas ponderadas também surge com possíveis aplicações para o método.

Quando considera-se as normas L_1 e L_∞ o problema pode ser formulado como programação linear (veja por exemplo (PEDROSO e MURATA, 2001)). Assim, pretende-se desenvolver formulações em programação linear para o problema de regressão ortogonal.

Por fim, outro ponto de interesse é o estudo de outras formas de regularização para o problema de regressão ortogonal. Uma abordagem nesse sentido foi apresentada recentemente por (LAMPEA e VOSS, 2013), usando a regularização de Tikhonov (TIKHONOV e ARSENIN, 1977). Este e outros tipos de regularização podem se tornar importantes do ponto de vista da regressão ortogonal com kernel principalmente quando trabalha-se com dados reais.

REFERÊNCIAS

ADCOCK, R. Note on the method of least squares. **Analyst 4**, p. 183–184, 1877.

AIZERMAN, M.; BRAVERMAN, E.; ROZONOER, L. Theoretical foundations of the potential function method in pattern recognition learning. **Automation and Remote Control**, v. 25, p. 821–837, 1964.

AKHIEZER, N.; GLAZMAN, I. **Theory of Linear Operators in Hilbert Spaces**, 1993.

ALCALá-FDEZ, J.; FERNANDEZ, A.; LUENGO, J.; DERRAC, J.; GARCíA, S.; SáNCHEZ, L.; HERRERA, F. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. **Journal of Multiple-Valued Logic and Soft Computing**, v. 17, p. 255–287, 2011.

AMMANN, L.; NESS, J. V. A routine for converting regression algorithms into corresponding orthogonal regression algorithms. **ACM Transactions on Mathematical Software**, v. 14, p. 76–87, 1988.

ARONSZAJN, N. Theory of reproducing kernels. **Transactions of the American Mathematical Society**, v. 68, p. 337–404, 1950.

BENNETT, K. P.; MANGASARIAN, O. L. Robust Linear Programming Discrimination of Two Linearly Inseparable Sets. **Optimization Methods and Software**, v. 1, p. 23–34, 1992.

BI, J.; BENNET, K. A geometric approach to support vector regression. **Neurocomputing**, v. 55, p. 79–108, 2003.

BOSER, B.; GUYON, I.; VAPNIK, V. A training algorithm for optimal margin classifiers. In: **Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory**, 1992. p. 144–152.

BRANHAM, R. Multivariate orthogonal regression in astronomy. **Celestial mechanics & dynamical astronomy**, v. 61, p. 239–251, 1995.

CAMPBELL, C. Kernel methods: A survey of current techniques. **Neurocomputing**, v. 48, p. 63–84, 2002.

CAMPS-VALLS, G.; BRUZZONE, L.; ROJO-ÁLVAREZ, J. L.; MELGANI, F. Robust Support Vector Regression for Biophysical Variable Estimation from Remotely Sensed Images. **IEEE Geoscience and Remote Sensing Letters**, v. 3, 2006.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, p. 273–297, 1995.

CRAMMER, K.; DEKEL, O.; KESHET, J.; SHALEV-SHWARTZ, S.; SINGER, Y. Online passive-aggressive algorithms. **Journal of Machine Learning Research**, v. 7, p. 551–585, 2006.

DAX, A. The distance between two convex sets. **Linear Algebra and its Applications**, v. 416, p. 184–213, 2006.

FILHO, F. F. C. **Algoritmos Numéricos**, 2007.

FRANK, A.; ASUNCION, A. **UCI Machine Learning Repository**. 2010. Disponível em: <http://archive.ics.uci.edu/ml>.

GENTILE, C. A new approximate maximal margin classification algorithm. **Journal of Machine Learning Research**, v. 2, p. 213–242, 2001.

GOLUB, G. H. Some Modified Matrix Eigenvalue Problems. **SIAM Review**, v. 15, p. 318–334, 1973.

GOLUB, G. H.; LOAN, C. F. V. **Matrix computations (3rd ed.)**, 1996.

GOLUB, G. H.; LOAN, C. V. An analysis of the total least squares problem. **SIAM J. Numer. Anal.**, v. 17, p. 883–893, 1980.

GRILICHES, Z.; RINGSTAD, V. Error-in-the-variables bias in nonlinear contexts. **Econometrica**, v. 38, n. 2, p. pp. 368–370, 1970.

HERBRICH, R. **Learning Kernel Classifiers: Theory and Algorithms**, 2002.

HERMUS, K.; VERHELST, W.; LEMMERLING, P.; WAMBACQ, P.; HUFFEL, S. V. Perceptual audio modeling with exponentially damped sinusoids. **Signal Processing**, v. 85, n. 1, p. 163 – 176, 2005.

HIRAKAWA, K.; PARKS, T. W. Image denoising using total least squares. **IEEE Transactions on Image Processing**, v. 15, p. 2730–2742, 2006.

HUBER, P. J. Robust statistics: A review. **The Annals of Mathematical Statistics**, v. 43, p. 1041–1067, 1972.

HUFFEL, S. V. **Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling**, 1997.

HUFFEL, S. V.; LEMMERLING, P. **Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications**, 2002.

HUFFEL, S. V.; VANDEWALLE, J. **The Total Least Squares Problem: Computational Aspects and Analysis**, 1991.

HUNTER, J. K.; NACHTERGAELE, B. **Applied Analysis**, 2001.

JOACHIMS, T. Making large-scale support vector machine learning practical. In: SCHöLKOPF, B.; BURGES, C.; SMOLA, A. (Ed.). **Advances in kernel methods**, 1999. p. 169–184.

KIMELDORF, G. S.; WAHBA, G. Some results on tchebycheffian spline funtions. **Journal of Mathematical Analysis and Applications**, v. 33, p. 82–95, 1971.

KIVINEN, J.; SMOLA, A.; WILLIAMSON, R. Online learning with kernels. **IEEE Transactions on Signal Processing**, v. 52, p. 2165–2176, 2004.

KUHN, H.; TUCKER, A. Nonlinear programming. In: **Proceedings, Second Berkeley Symposium on Mathematical Statistics and Probabilistics**, 1951. p. 481–492.

LAMPEA, J.; VOSS, H. Large-scale tikhonov regularization of total least squares. **Journal of Computational and Applied Mathematics**, v. 238, p. 95–108, 2013.

LEITE, S. C.; NETO, R. F. Incremental margin algorithm for large margin classifiers. **Neurocomputing**, v. 71, p. 1550–1560, 2008.

LI, Y.; LONG, P. M. The relaxed online maximum margin algorithm. **Machine Learning**, v. 46, p. 361–387, 2002.

LUONG, H. Q.; GOOSSENS, B.; PIZURICA, A.; PHILIPS, W. Joint photometric and geometric image registration in the total least square sense. **Pattern Recognition Letters**, v. 32, p. 2061–2067, 2011.

LUONG, H. Q.; GOOSSENS, B.; PIZURICA, A.; PHILIPS, W. Total least square kernel regression. **Journal of Visual Communication and Image Representation**, v. 23, p. 94–99, 2012.

MARKOVSKY, I.; HUFFEL, S. V. Overview of total least-square methods. **Signal Processing**, v. 87, p. 2283–2303, 2007.

MARKOVSKY(2010), I. Bibliography on total least squares and related methods. **Statistics and Its Interface**, v. 3, p. 329–334, 2010.

MERCER, J. Functions of positive and negative type, and their connection with the theory of integral equations. **Philosophical Transactions of the Royal Society**, v. 209, p. 415–446, 1909.

MÜHLICH, M.; MESTERLKOPF, R. The role of total least squares in motion analysis. In: **Proceedings, Fifth European Conference on Computer Vision**, 1998. p. 305–321.

NOVIKOFF, A. B. On convergence proofs for perceptrons. In: **Proceedings of the Symposium on the Mathematical Theory of Automata**, 1963. v. 12, p. 615–622.

PEDROSO, J. P.; MURATA, N. Support vector machines with different norms: motivation, formulations and results. **Pattern Recognition Letters**, v. 22, p. 1263–1272, 2001.

RAWLINGS, J. O.; PANTULA, S. G.; DICKEY, D. A. **Applied Regression Analysis: A Research Tool**, 1998.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, p. 386–408, 1958.

SCHÖLKOPF, B.; BARTLETT, P.; SMOLA, A.; WILLIAMSON, R. Support vector regression with automatic accuracy control. In: **Proceedings of ICANN'98, Perspectives in Neural Computing**, 1998. p. 111–116.

SCHUERMANS, M.; MARKOVSKY, I.; WENTZELL, P. D.; HUFFEL, S. V. On the equivalence between total least squares and maximum likelihood pca. **Analytica Chimica Acta**, v. 544, p. 254–267, 2005.

SHALEV-SHWARTZ, S.; SINGER, Y.; SREBRO, N. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In: **Proceedings of the 24th international conference on Machine learning**, 2007. p. 807–814.

SMOLA, A.; SCHÖLKOPF, B. **Learning with Kernels**, 2002.

SMOLA, A. J.; SCHÖLKOPF, B. **A tutorial on support vector regression**. 1998. NeuroCOLT2 Technical Report NC2-TR-1998-030.

STRANG, G. The fundamental theorem of linear algebra. **The American Mathematical Monthly**, v. 100, p. 848–855, 1993.

SUYKENS, J. A. K.; VANDEWALLE, J. Least squares support vector machine classifiers. **Neural Processing Letters**, v. 9, p. 293–300, 1999.

TIKHONOV, A. N.; ARSENIN, V. I. A. **Solutions of ill-posed problems**, 1977.

VAPNIK, V.; LERNER, A. Pattern Recognition using Generalized Portrait Method. **Automation and Remote Control**, v. 24, 1963.

VAPNIK, V. N. **The Nature of Statistical Learning Theory**, 1995.

WATKINS, D. S. **Fundamentals of Matrix Computations**, 2002.