Solução de Regressão Regularizada com Vetores Suporte através de Programação Linear

Lucas A. Teixeira[1], **Raul F. Neto**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora – Juiz de Fora, MG – Brasil

lucas.almeida1@ice.ufjf.br, raulfonseca.neto@ufjf.edu.br

Abstract. *In this work two regression methods based on support vector theory were introduced. These methods aim to find sparse solutions and are solvable by linear programming. One of them is only applicable to linear regression however the other can be extended to the nonlinear case through kernel methods. The proposed methods obtained numerical results close to state of the art methods.*

Resumo. *Nesse trabalho foram introduzidos dois métodos de regressão baseados na teoria de vetor suporte que visam encontrar soluções esparsas. Esses métodos são solucionáveis por programação linear. Um deles só aplicável a regressão linear entretanto o outro pode ser estendido para o caso não-linear através de métodos kernel. Os métodos propostos obtiveram resultados numéricos próximos dos métodos considerados estado da arte.*

1. Introdução

Regressão é um processo estatístico usado para encontrar a relação de uma variável, chamada variável dependente, com uma ou mais variáveis, chamadas variáveis independentes. A regressão tem uso nas mais diversas áreas, como por exemplo: física [Boucher et al. 2015], engenharia [De Rivas et al. 2017], biologia [Alves et al. 2019] e etc. No aprendizado de máquinas são estudados métodos que encontram uma função (função de regressão) que explícita a relação das variáveis independentes com a variável dependente. Estes métodos dependem de um conjunto de dados contendo amostras das variáveis e, portanto, fazem parte do aprendizado supervisionado.

Por partir de um conjunto de amostras e chegar na relação entre as variáveis a regressão é um problema inverso por definição e por isso resulta na maioria das vezes em problemas malpostos. Isso significa que algoritmos de regressão estão sujeitos ao sobreajuste (*overfitting*), i.e., um ajuste tão preciso nos dados – e em pequenos erros inerentes aos dados – que torna a função de regressão tão complexa a ponto de impossibilitar seu uso para interpretação dos dados ou predição de novas amostras. Esse problema pode ser contornado por diferentes abordagens, sendo a regularização a mais comum entre elas pois consiste em penalizar a complexidade da função de regressão. Outra abordagem para diminuir o sobreajuste é a seleção de características que elimina algumas variáveis independentes que não tem relação com a variável dependente, evitando que qualquer variância destas variáveis confunda o algoritmo. Outro efeito positivo da seleção de características é a atenuação dos efeitos da maldição da dimensionalidade – um problema

que ocorre em dimensões elevadas pois exige cada vez mais amostras para que o conjunto de dados seja representativo e mais poder computacional para processar tantas amostras.

Neste trabalho são propostos dois modelos de programação linear baseados em vetores suporte que aplicam regularização e seleção de característica através de uma nova abordagem que restringe a norma ℓ_1 do vetor de parâmetros.

Este trabalho foi dividido nas seguintes seções: a Seção 2 introduz conceitos abordados neste artigo e apresenta outros trabalhos relacionados; na Seção 3 são propostos os modelos de programação linear; na Seção 4 são discutidos os experimentos, as medidas de desempenho e os conjuntos de dados utilizados; na Seção 5 são analisados os resultados obtidos por tais experimentos; por fim, a Seção 6 encerra o trabalho com as principais observações e indicação de trabalhos futuros.

2. Fundamentação Teórica

Esta seção contém uma definição formal do problema de regressão e apresenta importantes conceitos utilizados em sua solução ao mesmo tempo que discute os algoritmos que introduziram tais conceitos. Ao final são discutidos outros trabalhos também relevantes.

De acordo com a teoria da aprendizagem estatística o problema da regressão pode ser formalmente definido da seguinte maneira: a partir do conjunto de dados $\{\mathbf{x_i}, y_i\}_{i=1}^{n}$ – no qual $x_i \in \mathbb{R}^d$ é um vetor com as variáveis independentes e $y_i \in \mathbb{R}$ é a variável dependente para cada amostra i – encontrar uma função $h(\mathbf{x})$ que minimize uma função de perda $L : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ – a função de perda relaciona os valores estimados ($\hat{\mathbf{y}} = h(\mathbf{x_i}), \forall i = 0, \ldots, n$) e os valores reais $\mathbf{y}$ com um custo definido pela função.

O primeiro algoritmo que resolve esse tipo de problema é o método dos mínimos quadrados (*least squares*). Assim como em outros métodos paramétricos a função $h(\mathbf{x})$ é substituída por uma função f qualquer da forma $f(\mathbf{x}, \mathbf{w})$ e o problema transforma-se em procurar o vetor de parâmetros w que minimiza a função de perda. No caso do *least squares* a função de perda é o erro quadrático médio (*mean squared error* – MSE) expresso por:

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 . \tag{1}$$

ou usando a norma ℓ_2 – também conhecida como distância euclidiana e representada por $\|\cdot\|_2$ – e ignorando a divisão por n por se tratar uma constante, tem-se a expressão:

$$L_{MSE} = \|\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}\|_2^2 \tag{2}$$

Para o caso em que a função f é uma combinação linear das variáveis independentes é possível calcular $\mathbf{w}$, esse método é chamado mínimos quadrados ordinários (*ordinary least squares* ou OLS). Para tanto escreve-se o problema na forma matricial, usando os termos:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1d} \\ 1 & x_{21} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{nd} \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}, \tag{3}$$

onde w_0 é a interseção com o eixo y por isso uma coluna de 1 foi adicionada a $\mathbf{X}$; então a forma matricial é:

$$\mathbf{X}\mathbf{w} = \hat{\mathbf{y}}. \tag{4}$$

Substituindo essa forma na função de perda MSE (Equação 2) tem-se:

$$L_{MSE} = (\mathbf{y} - \mathbf{X}\mathbf{w})^\intercal (\mathbf{y} - \mathbf{X}\mathbf{w}) \tag{5}$$

que pode ser expandido na forma:

$$L_{MSE} = \mathbf{y}^\intercal\mathbf{y} - \mathbf{y}^\intercal\mathbf{X}\mathbf{w} - \mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{y} + \mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{X}\mathbf{w}. \tag{6}$$

Ao derivar a função de perda em relação a $\mathbf{w}$ e igualar a zero podemos encontrar um ponto crítico que devido a forma da função será um ponto de mínimo, ou seja, a solução para o problema de minimização. E então para quando $X^\intercal X$ é invertível tem-se a seguinte solução do OLS:

$$\mathbf{w} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y} \tag{7}$$

Para os casos em que $\mathbf{X}^\intercal\mathbf{X}$ não é invertível ou para evitar o sobreajuste foi proposta em [Tikhonov 1963] a regularização de Tikhonov (também conhecida como *ridge regression*). Este método é uma extensão do OLS e minimiza a mesma função de perda acrescida do termo de regularização que penaliza funções complexas. Como f é uma combinação linear das variáveis independentes sua complexidade depende dos parâmetros $\mathbf{w}$ exceto pelo w_0 que é a interseção com o eixo y, porém esse parâmetro pode ser ignorado se as amostras forem centralizadas na origem. Outro pré-processamento fundamental é a normalização para eliminar a escala das variáveis independentes. Considerando que ambos pré-processamentos foram realizados chega-se na seguinte equação:

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \|\lambda\mathbf{I}\mathbf{w}\|_2^2, \tag{8}$$

na qual $\mathbf{I}$ é uma matriz identidade de tamanho $d \times d$ e λ é um hiperparâmetro positivo que controla o balanço entre viés e variância. A norma ℓ_2 foi utilizada no termo de regularização por ser diferenciável e, portanto, consegue-se encontrar uma solução com forma fechada seguindo os mesmos passos do OLS. A seguinte solução não depende da invertibilidade da matriz $X^\intercal X$ e é única:

$$\mathbf{w} = \left(\mathbf{X}^\intercal\mathbf{X} + \lambda^2\mathbf{I}\right)^{-1}\mathbf{X}^\intercal\mathbf{y}. \tag{9}$$

Uma outra variante do OLS é o LASSO (*least absolute shrinkage and selection operator*) [Tibshirani 1996]. A principal característica do LASSO é que seu termo de

regularização contém a norma ℓ_1 – também conhecida como distância de Manhattan e representada por $\|\cdot\|_1$. A solução do LASSO é:

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_1 . \tag{10}$$

Devido aos diferentes termos de regularização existem diferenças entre o LASSO e a regularização de Tikhonov, as principais delas são: o LASSO não possui forma fechada pois a norma ℓ_1 não é diferenciável; o LASSO realiza regularização e seleção de características ao reduzir alguns parâmetros a zero; em casos em que existem mais características que amostras ($d > n$) o LASSO seleciona no máximo n características; em casos em que existe multicolinearidade o LASSO pode ter um desempenho pior que a regularização de Tikhonov porque pode reduzir parâmetros importantes a zero desperdiçando informação relevante.

Para mesclar os pontos positivos da regressão de Tikhonov e do LASSO foi desenvolvida a *elastic net*[Zou and Hastie 2005]. Assim como o LASSO este algoritmo faz seleção de características e não possui forma fechada porém contorna os outros problemas do LASSO citados anteriormente. Sua equação é:

$$\min_{\mathbf{w}} \left\{ \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + (1-\alpha)\|\mathbf{w}\|_1 + \alpha\|\mathbf{w}\|_2^2 \right\}, \tag{11}$$

no qual $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2}$, assim sendo α funciona como um balanço entre regularização de Tikhonov e o LASSO.

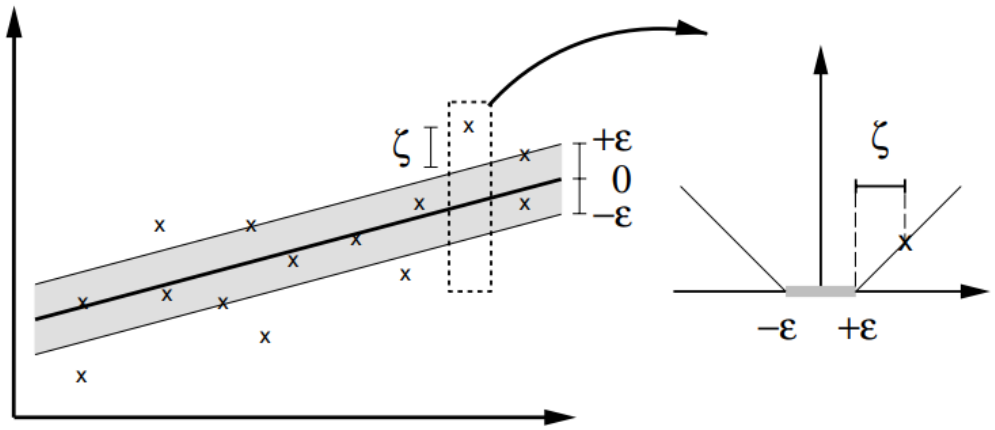

Figure 1. Exemplo da perda ε-insensível para uma função linear. Pontos na região cinza não contribuem com a solução. Fonte: [Smola and Schölkopf 2004].

Em [Drucker et al. 1997] foi proposta uma abordagem baseada em vetores suporte que ficou conhecida como *support vector regression* (SVR), essa abordagem é uma extensão das máquinas de vetor suporte (support vector machine – SVM) desenvolvidas em [Boser et al. 1992]. Esse método usa um termo de regularização igual a norma ℓ_2 e uma nova função de perda chamada de ε-insensível dada por:

$$L_\varepsilon = \sum_{i=1}^{n} max(0, |y_i - \hat{y}_i| - \varepsilon), \tag{12}$$

em que ε é o hiperparâmetro de insensibilidade e permite que sejam ignorados erros menores que seu valor (ver Figura 2). Apesar da adição desse hiperparâmetro, essa função de perda possui algumas vantagens como robustez a *outliers*, o efeito regulatório e soluções esparsas. Isso ocorre pois parte das amostras têm seu erro reduzido a zero e não afetam a solução, para as amostras que afetam a solução dá-se o nome de vetores suporte. A solução pode ser encontrada através de programação quadrática ao se resolver o seguinte problema dual:

$$\begin{aligned} \text{maximizar} \quad & -\frac{1}{2}\sum_{i,j=1}^{n}\left(\alpha_i-\alpha_i^*\right)\left(\alpha_i-\alpha_i^*\right)\mathbf{x_i}^\top\mathbf{x_j} \\ & -\varepsilon\sum_{i=1}^{n}\left(\alpha_i+\alpha_i^*\right)+\sum_{i=1}^{n}\left(\alpha_i-\alpha_i^*\right)y_i \qquad (13a) \\ \text{sujeito a} \quad & \sum_{i=1}^{n}\left(\alpha_i^*-\alpha_i\right)=0 \qquad (13b) \\ & 0\leq\alpha_i,\alpha_i^*\leq C \qquad (13c) \end{aligned}$$

no qual C é o hiperparâmetro que controla a relação entre bias e variância. A representação dessa solução é:

$$\mathbf{w}=\sum_{i=1}^{n}\left(\alpha_i-\alpha_i^*\right)\mathbf{x_i} \qquad (14)$$

ou representada diretamente por:

$$h(\mathbf{x})=f(\mathbf{x},\mathbf{w})=\sum_{i=1}^{n}\left(\alpha_i-\alpha_i^*\right)\mathbf{x_i}^\top\mathbf{x_j}+b \qquad (15)$$

Além das vantagens já citadas esse método também pode aplicar o *kernel trick* para aproximar funções não-lineares. O *kernel trick* permite o mapeamento indireto das variáveis independentes em um espaço de características usando um *kernel* k. Ao resolver a aproximação linear no espaço de características, encontra-se uma função não-linear em termos das variáveis independentes. Para uma função ser um *kernel* ela deve ser contínua e obedecer as condições de Mercer[Mercer 1909] de simetria (Equação 16) e positividade semi-definida (Equação 17) para todas as funções quadraticamente integráveis $g(x)$.

$$k(\mathbf{x},\mathbf{x}')=k(\mathbf{x}',\mathbf{x}) \qquad (16)$$

$$\iint g(\mathbf{x})k(\mathbf{x},\mathbf{x}')g(\mathbf{x}')\,d\mathbf{x}\,d\mathbf{x}'\geq 0 \qquad (17)$$

O *kernel* usado nesse trabalho foi o *kernel* gaussiano dado por:

$$k_{rbf}(\mathbf{x},\mathbf{x}')=e^{-\gamma\|\mathbf{x}-\mathbf{x}'\|^2}, \qquad (18)$$

onde γ é um hiperparâmetro.

Um outro método baseado em vetores suporte foi proposto em [Smola et al. 1999] com objetivo de encontrar soluções esparsas. Esse método penaliza a complexidade por adicionar um termo de regularização com a norma ℓ_1. A solução desse método pode ser encontrada ao resolver o seguinte problema de programação linear usando as variáveis duais:

$$\text{minimizar} \quad \sum_{i=1}^{n} \left(\alpha_j + \alpha_j^*\right) + C \sum_{i=1}^{n} \left(\xi_i + \xi_i^*\right) \tag{19a}$$

$$\text{sujeito a} \quad y_i - \sum_{j=1}^{n} \left(\alpha_j - \alpha_j^*\right) k\left(\mathbf{x_i}, \mathbf{x_j}\right) - b \leq \varepsilon + \xi_i \tag{19b}$$

$$\sum_{j=1}^{n} \left(\alpha_j - \alpha_j^*\right) k\left(\mathbf{x_j}, \mathbf{x_i}\right) + b - y_i \leq \varepsilon + \xi_i^* \tag{19c}$$

$$\alpha_i, \alpha_i^*, \xi_i, \xi_i^* \geq 0 \tag{19d}$$

Esse método também foi utilizado em [Bi et al. 2003] como parte de um algoritmo para realizar seleção de características para regressão linear.

Em [Jaggi 2013] foi provada a equivalência entre o LASSO e o SVM e depois em [Zhou et al. 2015] foi feita a redução da *elastic net* para o SVM. Esses resultados viabilizam a solução de problemas desses métodos – que antes só podiam ser resolvidas por métodos de subgradiente – por algoritmos otimizados para SVM.

A Tabela 1 resume as características de cada método discutido até agora.

Table 1. Comparações dos métodos apresentados.

Método	**Função de perda**	**Regularização**
OLS	ℓ_2	–
Regularização de Tikhonov	ℓ_2	ℓ_2
LASSO	ℓ_2	ℓ_1
Elastic net	ℓ_2	ℓ_1 e ℓ_2
SVR	ε-insensível	ℓ_2
SVR-LP	ε-insensível	ℓ_1

3. Proposta

Nesta seção é apresentada uma variação do SVR-LP com o objetivo de facilitar o controle da esparsidade da solução. Ao invés de adicionar um termo de regularização usa-se uma restrição da norma ℓ_1 do vetor de parâmetros, dessa forma tem-se:

$$\min_{\mathbf{w},b} L_\varepsilon(\mathbf{y}, \mathbf{X}\mathbf{w} + b) \quad \text{sujeito a} \quad |\mathbf{w}|_1 \leq t \tag{20}$$

ou usando utilizando um *kernel* k para funções não lineares:

$$\min_{\boldsymbol{\alpha},b} L_\varepsilon(\mathbf{y}, \boldsymbol{\alpha}\mathbf{K} + b) \quad \text{sujeito a} \quad |\boldsymbol{\alpha}|_1 \leq t \tag{21}$$

no qual $\mathbf{K}$ é uma matriz tal que $\mathbf{K}_{i,j} = k(x_i, x_j)$ e $t \geq 0$ é um hiperparâmetro de controle da esparsidade. Para os casos em que $t = 0$ todos os parâmetros serão zero e aumentar

t faz com que a norma do vetor de parâmetros aumente e gradualmente os parâmetros deixam de ser zero. A solução da Equação 21 usando variáveis duais é a solução do seguinte modelo de programação linear:

$$\text{minimizar} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \tag{22a}$$

$$\text{sujeito a} \quad y_i - \sum_{j=1}^{n}\left(\alpha_j - \alpha_j^*\right) k\left(\mathbf{x_j}, \mathbf{x_i}\right) - b \leq \varepsilon + \xi_i \tag{22b}$$

$$\sum_{j=1}^{n}\left(\alpha_j - \alpha_j^*\right) k\left(\mathbf{x_j}, \mathbf{x_i}\right) + b - y_i \leq \varepsilon + \xi_i^* \tag{22c}$$

$$\sum_{i=1}^{n}\left(\alpha_i + \alpha_i^*\right) \leq t \tag{22d}$$

$$\alpha_i, \alpha_i^*, \xi_i, \xi_i^* \geq 0 \tag{22e}$$

Na qual a restrição descrita na Equação 22d é a restrição da norma ℓ_1 do vetor de parâmetros. Esse modelo é apresentado como uma variação do SVR-LP, pois é possível aplicar um multiplicador de Lagrange na restrição 22d e obter o SVR-LP.

Para o caso linear também é proposta uma penalização dos parâmetros em função da correlação entre a variável independente associada ao parâmetro e a variável dependente. Essa proposta visa restringir a influência de variáveis pouco significativas e aumentar a esparsidade da solução. Assim sendo quanto menor a correlação entre as variáveis maior deve ser a penalidade. Usando o coeficiente de correlação de Pearson (Equação 23) que varia entre $[-1, 1]$, sendo 0 o valor quando não existe dependência linear, podemos criar uma penalização ao dividir o parâmetro pelo valor absoluto do coeficiente de correlação.

$$cor_{xy} = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}} \tag{23}$$

Então, definindo cor_i como a correlação entre a i-ésima variável independente e y, podemos encontrar o resultado para funções lineares – pois o coeficiente de Pearson indica apenas dependência linear – resolvendo o modelo com variáveis primais:

$$\text{minimizar} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \tag{24a}$$

$$\text{sujeito a} \quad y_i - \left\langle \mathbf{w}^+ - \mathbf{w}^-, \mathbf{x_i} \right\rangle - b \leq \varepsilon + \xi_i \tag{24b}$$

$$\left\langle \mathbf{w}^+ - \mathbf{w}^-, \mathbf{x_i} \right\rangle + b - y_i \leq \varepsilon + \xi_i^* \tag{24c}$$

$$\sum_{i=1}^{d}\frac{\left(w_i^+ + w_i^-\right)}{|cor_i|} \leq t \tag{24d}$$

$$w_i^+, w_i^-, \xi_i, \xi_i^* \geq 0 \tag{24e}$$

O modelo apresentado em 22 é chamado de RSVR (*restricted support vector regression*) e o modelo descrito em 24 é chamado PRSVR (*penalized restricted support vector regression*) pois a Equação 24d é a forma penalizada da restrição descrita na Equação 20.

4. Metodologia

Nesta seção são descritos experimentos e métricas usados para comparar os modelos de otimização desenvolvidos na Seção 3 com alguns algoritmos apresentados na Seção 2.

Qualquer função de perda pode ser usada para comparar as diferentes implementações, em particular o MSE (Equação 2) será utilizado neste trabalho.

$$L_{MSE} = \|\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}\|_2^2 \qquad \text{(2 reapresentada)}$$

Além do MSE, também será usado o erro absoluto médio (*mean absolute error* – MAE) cujo valor é:

$$L_{MAE} = |y_i - \hat{y}_i| \,. \tag{25}$$

Para regressão linear também é comum o uso do coeficiente de determinação dado pela seguinte formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2}, \tag{26}$$

no qual $\overline{y}$ é a media de $\mathbf{y}$. Esse coeficiente significa o quanto o modelo prediz a variável dependente, o valor máximo desse coeficiente é 1 e só é atingido quando o modelo prediz por completo a variável dependente.

A Tabela 4 apresenta os conjuntos de dados. A base *boston housing* foi obtida no repositório de aprendizado de máquinas da UCI[Dua and Graff 2017]. As demais bases são artificiais e os detalhes de como foram geradas está descrito a seguir.

Table 2. Conjunto de dados utilizados neste trabalho, n é o número de amostras e d é o número de variáveis independentes.

Base	n	d
Linear	120	20
LinearCov	120	20
Boston housing	506	13
Sinc	100	1
Exp	156	1

A base Linear foi criada a partir de um vetor com 20 variáveis aleatórias $\mathbf{x}$ que são independentes e identicamente distribuídas. Essas variáveis foram usadas para calcular a variável dependente da seguinte forma: $y = 5x_1 + x_2 + x_3 + x_4 + x_5 - 10x_{20}$ e, por fim, y foi poluída acrescentando um ruído gaussiano cuja média e desvio padrão são $\mu = 0, \sigma = 4$. A base LinearCov foi gerada pelo mesmo processo entretanto ao invés das variáveis $\mathbf{x}$ serem i.i.d existe covariância entre as variáveis, tal que $cov(x_i, x_j) = 0,8^{|i-j|}$.

As bases *Sinc* e *Exp* foram criadas a partir de funções não lineares acrescidas de ruído gaussiano. A *Sinc* foi gerada do seguinte modo: para a variável aleatória $x \in [-10, 10]$, $y = sinc(x) = sin(x)/x$, e em seguida é acrescentado ruído gaussiano ($\mu = 0, \sigma = 0.2$). De modo similar *Exp* foi gerada a partir da função $y = e^{-x^2}$ para $x \in [-1, 1]$ e adicionando o ruído ($\mu = 0, \sigma = 0.3$).

Os métodos propostos foram comparados aos seguintes métodos: regularização de Tikhonov, LASSO, *elastic net*, SVR e SVR-LP. Os primeiros três métodos foram ajustados usando a biblioteca *glmnet* – que implementa a solução expressa em [Friedman et al. 2010]. Para solução da SVR foi feita usando a *libSVM*[Chang and Lin 2011] através do pacote *e1071*[Meyer et al. 2017] enquanto a SVR-LP e os modelos propostos foram resolvidos usando o CPLEX versão 12.8 através da biblioteca *cplexAPI*[Gelius-Dietrich 2017].

Inicialmente fez-se um experimento de regressão linear nas bases Linear, LinearCov *boston housing* usando os métodos citados anteriormente e o PRSVR. Ambas SVR e SVR-LP usaram o *kernel* linear. Depois foi feito um experimento de regressão não-linear nas bases *boston housing*, *Sinc* e *Exp* com os métodos SVR, SVR-LP e RSVR, todos usando o *kernel* gaussiano. Em ambos experimentos foram feitos 10 testes usando dois terços do conjunto de dados para treino e um terço para teste. Uma busca por hiperparâmetros através da validação cruzada também foi feita para ambos experimentos.

A validação cruzada foi um 10-*fold* aplicado ao conjunto de treino, escolhendo os hiperparâmetros que resultavam em menor MSE. O *glmnet* faz a seleção do parâmetro λ de forma automática então apenas o parâmetro α da *elastic net* foi otimizado, os valores possíveis foram de $0,05$ a $0,95$ com intervalo de $0,05$. Para a SVR, SVR-LP os parâmetros ε, C e γ (apenas para o caso do *kernel* gaussiano) foi feita busca em *grid*, para os valores $\varepsilon = \{2^{-6}, 2^{-5}, \ldots, 2^{-1}\}, C = \{2^{-5}, 2^{-2}, \ldots, 2^{10}\}, \gamma = \{2^{-13}, 2^{-9}, \ldots, 2^{3}\}$. PRSVR e RSVR também realizaram busca em *grid*, o primeiro nos parâmetros ε e t e o segundo nos parâmetros ε, γ e t; com $t = \{2^2, 2^3, \ldots, 2^7\}$ em ambos casos.

Por fim, é feita uma analise da variação do hiperparâmetro t para os métodos propostos para mostrar o efeito dessa variação no vetor de parâmetros.

5. Resultados

Esta seção discute os resultados obtidos ao seguir a metodologia descrita na Seção 4.

5.1. Regressão linear

Os resultados estão apresentados na Tabela 3, os melhores resultados foram os obtidos pelo LASSO e a *elastic net*. Ao comparar os métodos SV entre si percebe-se que o PRSVR alcançou resultados melhores na base Linear e iguais nas demais. Conjectura-se que isso ocorreu devido ao fato da base Linear possuir amostras i.i.d. o que pode ter permitido que a correlação entre as variáveis independentes e a variável dependente pudesse ser observada nas amostras. Se este for o motivo o PRSVR não apresentou piora nos casos em que os pontos não são i.i.d., como no LinearCov.

5.2. Regressão não-linear

Os resultados estão apresentados na Tabela 4. No geral os métodos mais simples SVR-LP e RSVR ficaram equiparáveis à SVR enquanto mantinham a solução esparsa – observável

Table 3. Valores de MAE, MSE e R^2 calculados a partir da média de 10 testes.

	Linear			LinearCov			Housing		
	MAE	MSE	R^2	MAE	MSE	R^2	MAE	MSE	R^2
LASSO	**0.294**	**0.122**	**0.863**	0.277	0.105	0.883	0.372	**0.283**	**0.710**
RIDGE	0.324	0.146	0.837	0.289	0.115	0.874	**0.366**	0.286	0.709
ENET	0.295	0.123	0.862	**0.274**	**0.104**	**0.884**	0.371	**0.283**	**0.710**
SVR	0.331	0.154	0.829	0.295	0.122	0.866	0.374	0.287	0.705
SVR-LP	0.338	0.166	0.816	0.292	0.120	0.868	0.376	0.288	0.704
PRSVR	0.308	0.135	0.850	0.281	0.119	0.868	0.378	0.288	0.704

pelo número de vetores suporte. Para a base Exp essa esparsidade contribuiu para uma melhor aproximação para ambos. Essa tendência não se repetiu na base *boston housing* onde a SVR encontrou o melhor e o mais esparso resultado.

Table 4. Valores de MAE e MSE calculados a partir da média de 10 testes, #SV é o número de vetores suporte.

	Boston Housing			Sinc			Exp		
	MAE	MSE	#SV	MAE	MSE	#SV	MAE	MSE	#SV
SVR	**0.261**	**0.148**	**127.8**	**0.418**	**0.263**	51.6	0.663	0.725	83.9
SVR-LP	0.306	0.266	262.7	0.434	0.289	6.0	0.646	**0.334**	**2.0**
RSVR	0.309	0.269	212.9	0.437	0.295	**5.9**	**0.645**	0.683	6.6

5.3. Parâmetro t

Nas Figuras 2 e 3 pode-se ver a variação do número de vetores suporte com o aumento de t para o PRSVR e o RSVR respectivamente. Observe que para valores altos de t o número de vetores suporte começa a se estabilizar, como ocorre na Figura 2 onde se chega ao máximo. Isso também pode ser observado através da escala da variável t nas figuras e a curva da Figura 3, na média cada vetor suporte precisa de um incremento maior em t do que o anterior.

6. Conclusão

Neste trabalho foram propostos dois métodos o PRSVR e o RSVR para regressão linear e não-linear respectivamente. Esses métodos podem ser solucionados por programação linear e mantém a esparsidade do vetor de parâmetros.

Deve-se levar em consideração que tanto o SVR-LP quanto os métodos propostos são solucionáveis através de programação linear um algoritmo menos complexo que as soluções da SVR, o LASSO e a *elastic net* e mesmo assim atingiram resultados próximos na maioria dos testes.

O modelo PRSVR obteve resultados competitivos com os outros métodos de vetores suporte, porém não ficou determinada a razão então novos trabalhos podem ser feitos nessa linha. Os métodos também podem vir a ser aplicados em mais conjuntos de dados no futuro.

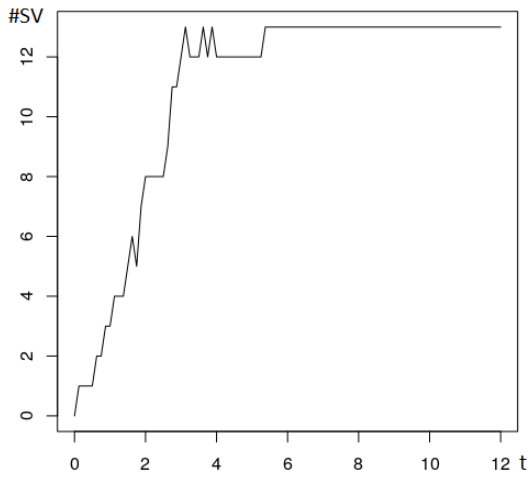

Figure 2. PRSVR na *Boston Housing*.

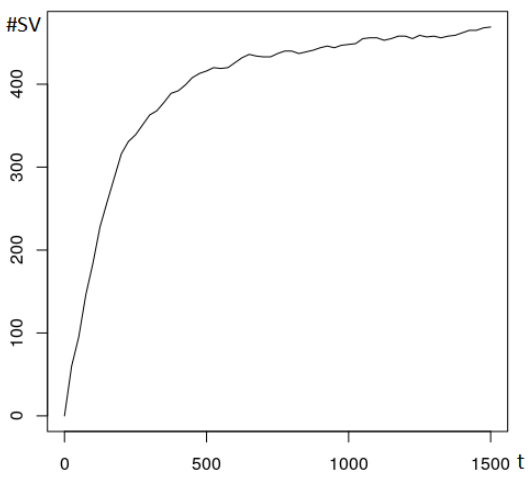

Figure 3. RSVR na *Boston Housing*.

7. Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

References

Alves, A., Chaparro Pinzon, A., Costa, R., Silva, M., Vieira, E., Mendonça, I., Viana, V., and Lôbo, R. (2019). Multiple regression and machine learning based methods for carcass traits and saleable meat cuts prediction using non-invasive in vivo measurements in commercial lambs. *Small Ruminant Research*, 171:49–56. cited By 0.

Bi, J., Bennett, K., Embrechts, M., Breneman, C., and Song, M. (2003). Dimensionality reduction via sparse support vector machines. *J. Mach. Learn. Res.*, 3:1229–1243.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Boucher, T., Ozanne, M., Carmosino, M., Dyar, M., Mahadevan, S., Breves, E., Lepore, K., and Clegg, S. (2015). A study of machine learning regression methods for major elemental analysis of rocks using laser-induced breakdown spectroscopy. *Spectrochimica Acta - Part B Atomic Spectroscopy*, 107:1–10. cited By 30.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

De Rivas, B., Vivancos, J.-L., Ordieres-Meré, J., and Capuz-Rizo, S. (2017). Determination of the total acid number (tan) of used mineral oils in aviation engines by ftir using regression models. *Chemometrics and Intelligent Laboratory Systems*, 160:32–39. cited By 6.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 155–161. MIT Press.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Gelius-Dietrich, G. (2017). *cplexAPI: R Interface to C API of IBM ILOG CPLEX*. R package version 1.3.3.

Jaggi, M. (2013). An equivalence between the lasso and support vector machines. *Regularization, optimization, kernels, and support vector machines, chap 1*.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2017). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-8.

Smola, A., Scholkopf, B., and Ratsch, G. (1999). Linear programs for automatic accuracy control in regression. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 575–580 vol.2.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tikhonov, A. N. (1963). On the solution of ill-posed problems and the method of regularization. *Doklady Akademii Nauk SSSR*, 151(3):501–504.

Zhou, Q., Chen, W., Song, S., Gardner, J. R., Weinberger, K. Q., and Chen, Y. (2015). A reduction of the elastic net to support vector machines with an application to gpu computing. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3210–3216. AAAI Press.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.