UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA

CARINA BRUNEHILDE PINTO DA SILVA

A TÉCNICA LASSO E SUAS POTENCIALIDADES NA SELEÇÃO DE VARIÁVEIS PARA MODELOS LINEARES

FORTALEZA
2018

CARINA BRUNEHILDE PINTO DA SILVA

A TÉCNICA LASSO E SUAS POTENCIALIDADES NA SELEÇÃO DE VARIÁVEIS PARA MODELOS LINEARES

Monografia submetida ao curso de Bacharelado em Estatística do Departamento de Estatística e Matemática Aplicada da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Bacharel em Estatística.

Orientador: Prof. Dr. Juvêncio Santos Nobre

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S579t Silva, Carina Brunehilde Pinto da.
A técnica LASSO e suas potencialidades na seleção de variáveis para modelos lineares / Carina Brunehilde Pinto da Silva. – 2017.
60 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Ciências, Curso de Estatística, Fortaleza, 2017.
Orientação: Prof. Dr. Juvêncio Santos Nobre.

1. LASSO. 2. Modelos de regressão. 3. Seleção de variáveis. 4. Regressão L1. 5. Penalização. I. Título.
CDD 519.5

CARINA BRUNEHILDE PINTO DA SILVA

A TÉCNICA LASSO E SUAS POTENCIALIDADES NA SELEÇÃO DE VARIÁVEIS PARA MODELOS LINEARES

Monografia submetida ao curso de Bacharelado em Estatística do Departamento de Estatística e Matemática Aplicada da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Bacharel em Estatística.

Aprovada em: __ / __ / 2018.

BANCA EXAMINADORA

Prof. Dr. Juvêncio Santos Nobre (Orientador)
Universidade Federal do Ceará (UFC)

Profª. Dra. Maria Jacqueline Batista
Universidade Federal do Ceará (UFC)

Profª. Dra. Sílvia Maria de Freitas
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

À uma força superior, seja Deus, Buda, Allah, Tupã, Olodumaré, Zeus, anjo da guarda ou até mesmo a sorte, que sempre encaminha meus passos no caminho do bem e dos bons.

À minha família, especialmente aos meus avós, por serem porto seguro e fortaleza.

À Patrícia, que conhece mais do que ninguém cada lágrima que foi derrubada para que mais esse passo fosse dado em minha vida.

Ao meu orientador, Juvêncio Santos Nobre, pela confiança, pelas palavras de conforto, por dividir comigo um pouco do seu vasto conhecimento e por ser um exemplo da professora que um dia quero me tornar.

Às professoras Dra Maria Jacqueline Batista e Dra Sílvia Maria Freitas pela disponibilidade em participar desta banca e pelas correções sugeridas.

A todos os professores do DEMA, que foram tão compreensivos e incentivadores. O clima deste departamento é muito agradável e os aprendizados que tive aqui são incomensuráveis.

Aos colegas e amigos que fiz durante o curso.

À UFC, instituição a qual devo toda minha formação acadêmica.

Aos meus amigos, companheiros de vida, que dividem comigo todo fardo e toda glória.

Aos meus alunos, que também são motivação nesta minha busca por saber mais.

Obrigada! Nenhuma conquista é individual, e mesmo que fosse, não teria a menor graça. Afinal, para que serve a vida se não para ser dividida?

"We are drowning in information and starving for knowledge."

RESUMO

Esta pesquisa busca fazer um levantamento bibliográfico da técnica LASSO, *Least Absolute Selection and Shrinkage Operator*, bem como divulgar a produção de estudos brasileiros na área. A referida técnica mostra-se importante como uma alternativa para o ajuste de modelos com características específicas que dificultam a utilização da estimação via método de mínimos quadrados, para tanto, utiliza penalizações denominadas de L_1. Ao longo do trabalho utilizamos abordagens algébricas e geométricas para facilitar o entendimento do processo de estimação, tanto dos coeficientes dos modelos, como do parâmetro de ajuste da penalização. Dedicamos uma parte do trabalho para discutir algumas desvantagens do LASSO, bem como as alternativas e generalizações que estão surgindo para contorná-las. Encerramos as discussões apresentando o pacote `glmnet`, implementado no *software* `R` para ajustes usando o LASSO, através de dois exemplos, com os quais foi possível discorrer sobre as principais funções do pacote.

Palavras-chave: LASSO. Modelos de regressão. Seleção de variáveis. Regressão L_1. Penalização.

ABSTRACT

This research seeks to make a bibliographic survey of LASSO technique, Least Absolute Selection and Shrinkage Operator, as well as publicize the production of Brazilian studies in the area. This technique is important as an alternative for the adjustment of models with specific characteristics that make it difficult to use the estimation by least squares method, for which it uses penalties called L_1. Throughout the work we use algebraic and geometric approaches to facilitate the understanding of the estimation process of both the coefficients of the models and the tuning parameter. We devote most of the work to discussing some of the disadvantages of LASSO, as well as the alternatives and generalizations that are emerging to circumvent them. We conclude the discussions by presenting the *glmnet* package, implemented in R software for adjustments using lasso, through two examples, with which they were possible to discuss the main functions of the package.

Keywords: LASSO. Selection models. Selection of variables. Regression L_1. Penalty.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

O ajuste de modelos de regressão é um dos pilares da análise estatística. É comum que pesquisadores, a partir de um banco de dados, estejam envolvidos em estudar se há ou não relação entre um conjunto de variáveis, ditas explicativas, e como estas podem influenciar no comportamento de uma variável específica, comumente chamada de variável resposta. Também pode ser de interesse do pesquisador fazer previsões do comportamento desta variável resposta para valores das variáveis explicativas que são apenas hipotéticos. Essas perguntas podem ser respondidas a partir do ajuste de um modelo de regressão, o que obviamente não é uma tarefa simples e requer uma série de atenções.

Na situação descrita acima, estamos considerando que o analista já saiba previamente qual conjunto de variáveis explicativas é de fato significativo para sua análise, mas, na prática, raramente se dispõem desta informação. Assim, outra necessidade que surge é delimitar um conjunto das variáveis que vão ser consideradas no modelo, tendo sempre em mente que, quanto mais variáveis forem consideradas, mais informações teremos, em contrapartida maior será a variância dos estimadores, o que diminui sua precisão nas estimações. A prática descrita é chamada de seleção de variáveis e requer uma série de cuidados, bem como apresenta algumas restrições e limitações.

Neste contexto, no presente trabalho, discutiremos especialmente uma técnica estatística denominada LASSO, *Least Absolute Selection and Shrinkage Operator*, que foi proposta por Tibshirani(1996) como uma possibilidade para a estimação e o ajuste de modelo e seleção de variáveis em situações adversas, especialmente aquelas nas quais os populares estimadores de mínimos quadrados apresentam limitações. Desde então muitos pesquisadores vêm dedicando estudos ao melhoramento e à generalização do método. Neste trabalho, faremos um levantamento bibliográfico de algumas das pesquisas mais importantes publicadas sobre o tema, especificamente para modelos lineares.

Alguns de nossos suportes teóricos principais são: Tibshirani(1996), Zou *et al.*(2007), Friedman, Hastie e Tibshirani(2010), Montgomery *et al.*(2012), Murphy (2012), Hastie, Tibshirani e Wainwright(2015), Casagrande(2016) e Pereira(2017). Todos apresentam vastos estudos na referida área, evidenciando a importância do tema em questão.

A partir desta investigação, os objetivos deste estudo foram construídos, sendo estes: incentivar o desenvolvimento de pesquisas sobre o LASSO, uma técnica relativamente nova e com grande potencial de aplicação nas mais diversas áreas da Estatística; investigar as pesquisas que vem sendo produzidas sobre o assunto, brasileiras ou não, e aumentar o número de produções brasileiras nesta área, uma vez que a maioria das referências deste trabalho são internacionais.

Para tanto, apresentamos os aspectos teóricos envolvendo a estimação do modelo via LASSO, e para melhorar o entendimento, ao final do trabalho, apresentamos a

análise de dois conjuntos de dados, ambos analisados utilizando o *software* R Core Team (2017), escolhido como suporte computacional por sua disponibilidade gratuita e por sua inquestionável abrangência no meio acadêmico.

O primeiro conjunto de dados foi extraído de um trabalho de Thomas (1990 *apud* Hastie *et al.*, 2015) que busca ajustar um modelo para explicar e prever taxas de criminalidades em cidades dos Estados Unidos. A partir desse estudo, introduzimos os principais comandos do pacote `glmnet` para o ajuste de um modelo via LASSO.

Já o segundo exemplo foi baseado no banco de dados `Hitters`, disponível em James *et al.* (2013), que contém 19 variáveis e 322 observações e seu estudo objetivou construir um modelo capaz de predizer o salário de rebatedores que jogam na *Major League Baseball* (MLB). A partir dele, pudemos realizar uma comparação entre alguns métodos de seleção de variáveis, os quais dividimos em dois grupos: métodos *subset selection* e métodos *shrinkage*.

Quanto a divisão dos capítulos, a organização se deu de modo que no Capítulo 2 abordamos as características gerais do LASSO, elencamos os principais parâmetros a serem estimados e discutimos os ganhos da aplicação de penalizações nos métodos de estimação tradicionais. Já no Capítulo 3, realizamos uma breve explanação sobre o método de mínimos quadrados, destacando algumas de suas vantagens e desvantagens. O Capítulo 4 é inteiramente dedicado à discussão sobre as penalizações, trazendo uma comparação entre as técnicas LASSO e regressão em crista. As estimações dos parâmetros de regressão e de penalização foram discutidas no Capítulo 5. Já o Capítulo 6 foi dedicado a apresentar algumas generalizações do LASSO e a apresentar o pacote `glmnet`, disponível no *software* `R`, para o ajuste de modelos via LASSO. O Capítulo 7 finaliza essa pesquisa e apresentando algumas sugestões de pesquisas futuras.

Concluimos essa pesquisa bibliográfica introdutória com a certeza de que ainda há muito o que pesquisar e desenvolver sobre o LASSO, desde sua aplicação em outros tipos de modelo a técnicas mais avançadas de diagnóstico.

2 LASSO: CARACTERÍSTICAS GERAIS.

O uso dos estimadores de mínimos quadrados (MQ) para modelos de regressão linear já é consagrado pela literatura. Basicamente o método consiste em buscar o modelo que apresenta a menor distância euclidiana entre os valores observados e os valores ajustados. Estes estimadores possuem propriedades ótimas, destacando-se o teorema de Gauss-Markov, o qual garante que o estimador de mínimos quadrados possui a menor variância dentre todos os estimadores lineares não viciados, sendo assim amplamente utilizados em modelos de regressão. Porém, em algumas situações, os estimadores MQ podem não ser uma boa opção para a estimação dos modelos. Nestes casos, o método de estimação para modelos lineares chamado LASSO apresenta-se como uma interessante alternativa.

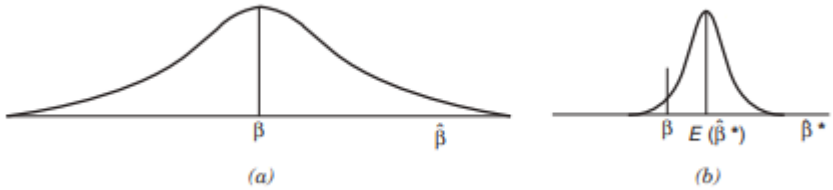
Proposto primeiramente por Robert Tibshirani no artigo *Regression Shrinkage and Selection via the LASSO* (1996), o LASSO, por suas características peculiares, pode ser considerado como uma possibilidade para selecionar variáveis ou como uma técnica de uso cada vez mais recorrente em problemas de Aprendizado de Máquina (*Machine Learning*) (MURPHY, 2012). Desde então, pesquisas sobre o método vem fomentando seu uso e ampliando ainda mais suas possibilidades de aplicação. Tibshirani(1996) aponta duas principais razões para que o analista busque outras possibilidades de estimação: precisão da previsão (*prediction accuracy*) e interpretação do modelo.

A melhora na acurácia da previsão se dá quando, mesmo estimadores não viciados, apresentam alta variância. Nesses casos acrescentar um viés em detrimento de uma variância menor pode ser conveniente, melhorando a precisão da previsão, ainda que produza estimadores viciados ou aumente o vício já existente. Montgomery *et al.*(2012) exemplifica este cenário através de duas situações representadas na Figura 1.

Em ambos os casos temos a distribuição amostral de estimadores para um parâmetro β, por exemplo. Na Figura 1 (a), $\hat{\beta}$ é um estimador não viciado para β, ou seja $\mathbb{E}(\hat{\beta}) = \beta$, porém $\hat{\beta}$ apresenta variância maior do que o estimador enviesado $\hat{\beta}^*$, representado na Figura 1 (b). A maior variação de $\hat{\beta}$ implica em intervalos de confiança com amplitudes maiores, diminuindo a precisão na estimação dos parâmetros. Flexibilizar a exigência de trabalhar com estimadores não viciados pode aliviar este problema, conforme discutiremos ao longo do trabalho. Já a melhora na interpretação se dá quando, através do LASSO, conseguimos escolher algumas das variáveis que apresentam efeitos mais significativos na variável resposta.

A seleção de variáveis é uma importante técnica estatística, pois possibilita que os analistas encontrem uma quantidade razoável de variáveis explicativas a serem utilizadas, de modo que proporcionem um bom balanceamento entre quantidade de informação aproveitada e variância das estimações. Outra vantagem da seleção de variáveis é a possibilidade de utilizá-la como correção para problemas de multicolinearidade. Sobre

Figura 1 – Exemplo de diferença entre a variância de estimadores viciados e não viciados.



Fonte: Montgomery et al. (2012)

a multicolinearidade, embora Montgomery et al.(2012) afirmem que a seleção de variáveis é a técnica mais comum a ser utilizada para tentar resolver estes problemas, de maneira geral, não existe garantia de que serão eliminados por completo. Problemas de multicolinearidade serão discutidos com mais profundidade nos capítulos futuros.

A busca por esse conjunto de variáveis pode ser um grande desafio para o analista, inclusive envolvendo grandes esforços computacionais. A seguir discutiremos algumas das técnicas mais clássicas utilizadas na realização desta escolha.

Inicialmente a seleção de variáveis pode se dar a partir da análise de todos os modelos que podem ser ajustados, ou seja, se dispomos de k variáveis explicativas, serão determinados todos os modelos com apenas uma delas, depois com duas e assim sucessivamente, o que nos leva a um total de $2^k - 1$ modelos a serem estudados. Para determinar o modelo a ser utilizado, deve-se fixar um critério de escolha. Vários são os critérios consagrados pela literatura para chancelar esta escolha, Montgomery et al. (2012) destacam: o coeficiente de determinação, o coeficiente de determinação ajustado, o quadrado médio dos resíduos, a estatística C_p de Mallows, o critério de Akaike (AIC) e sua proposta bayesiana (BIC).

Para os casos em que a análise de todos os modelos mostra-se muito onerosa ou inviável, foram desenvolvidos outros métodos de seleção de variáveis, que podem ser classificados em três grandes categorias: *forward selection*, *backward elimination* e *stepwise regression*. (MONTGOMERY et al., 2012)

De maneira sucinta, o método *forward selection*, cuja expressão traduzida significa "seleção para frente", consiste em considerar um modelo nulo, ou seja sem variáveis, e ir adicionando-as, uma a uma, até ser possível determinar o modelo a ser utilizado. A primeira variável a ser inserida será aquela que apresenta a maior correlação simples com a variável resposta, e o critério a ser aplicado será o valor da estatística F[1] para esta variável, que deve ser maior do que um valor F crítico. O modelo é ajustado com essa variável, e a partir daí usa-se novamente a variável explicativa que apresentar a maior correlação com a variável resposta a partir do novo modelo ajustado. O processo se re-

[1]Estatística calculada através da razão entre o Quadrado Médio da Regressão e o Quadrado Médio dos Resíduos, utilizada como parâmetro de seleção de variáveis em diversos testes de hipótese. Para maiores detalhes veja Montgomery et al.(2012)

pete até que a variável a ser testada não se encaixe mais no critério estipulado, então o processo para e o modelo é ajustado apenas com as variáveis que foram selecionadas.

Já o método *backward elimination*, traduzido seria "eliminação para trás", realiza a seleção no sentido inverso ao do método anterior. Neste caso, ajusta-se o modelo com todas as variáveis, e inicia-se um processo de eliminação daquelas que, a cada passo, apresentarem estatística F menor do que uma valor estipulado, começando pela variável que apresenta a menor correlação com a variável resposta.

Combinando características dos dois métodos anteriores, o *stepwise regression* adiciona e retira variáveis a partir do modelo nulo até conseguir determinar o modelo mais adequado para a descrição da situação de estudo.

Muitas são as críticas direcionadas às técnicas descritas, especialmente a de que não há grandes garantias de que as mesmas determinarão o melhor modelo, até porque, aplicadas ao mesmo banco de dados, podem apontar diferentes modelos como "os melhores". Na verdade, modelos são aproximações, sempre deve-se considerar o fato de que, por mais bem ajustado sejam, erros podem ser cometidos e nenhuma técnica poderá determiná-los com certeza. O ponto chave aqui é que, mesmo que o *forward selection*, o *backward elimination* e o *stepwise regression* sejam menos trabalhosos do que analisar todos os modelos possíveis, ainda sim, quanto maior for a quantidade de variáveis explicativas, mais difícil será sua utilização.

O LASSO surge então como uma interessante possibilidade para seleção de variáveis, uma vez que pode ser usado em análises com um grande banco de dados, especialmente se a quantidade de covariáveis for maior do que o número de observações, e ainda garante que uma boa parte dos coeficientes destas covariáveis seja nula, o que sugere que as demais são as características importantes a serem estudadas.

Esta característica é chamada de esparsidade (*sparsity*). Segundo Hastie et al. (2015), um modelo é dito esparso (*sparse model*) quando apenas alguns dos coeficientes possuem estimações diferentes de zero. Além de melhorar a interpretação, como já citamos anteriormente, modelos esparsos também possuem a vantagem de facilitar computacionalmente estas estimações.

Mas como o LASSO é capaz de possibilitar tais vantagens? A técnica minimiza a soma dos quadrados dos resíduos do modelo utilizando um parâmetro de ajuste s (*tuning parameter*), o qual deve ser maior do que a soma dos valores absolutos dos coeficientes do modelo, o que consiste em uma penalização do tipo L_1. Com isso vários coeficientes são "forçados" a zerar, daí vem a expressão *shrinkage* (do inglês, encolhimento). Os parâmetros não-nulos podem ser considerados os mais significativos na construção do modelo, assim a técnica também funciona para seleção de variáveis, daí o termo *selection* na sigla.

Assim, uma vez que o pesquisador suspeite que a utilização do LASSO na construção do modelo de interesse pode melhorar o ajuste, e pretenda utilizá-lo, terá de

resolver dois problemas iniciais, dos quais trataremos ao longo do trabalho: estimar os coeficientes do modelo e estimar o fator de penalização s.

No geral, chamamos de método de mínimos quadrados penalizados, todos os que se baseiam na distância entre valores observados e esperados, mas não utilizam a distância mínima. Destaca-se a utilização de duas técnicas, amplamente comparadas com o LASSO em trabalhos acadêmicos, são elas, *best-subset selection* e regressão em cristas.

A tradução literal de *best subset selection* é "seleção do melhor subconjunto", e é justamente isso que a técnica faz, selecionando o subconjunto de variáveis preditoras que ajustam o melhor modelo em termos de erro quadrático (HASTIE et al., 2017). Também conhecida como penalização L_0, pode ser um pouco complexa de desenvolver por se tratar de um problema não-convexo.

Já o estimador em crista submete o modelo a uma restrição do tipo L_2, ou seja, este é penalizado por uma constante que deve ser maior do que a soma dos quadrados dos coeficientes de regressão. O grande diferencial do LASSO com relação a regressão em crista é que a segunda aproxima os coeficientes de zero, mas não iguala-os a zero efetivamente, característica esta que o LASSO possui, transformando-o em uma interessante técnica de seleção de variáveis.

Tibshirani(1996) ressalta também a importância do método *non-negative garrote* como motivação inicial para o desenvolvimento da técnica LASSO. O estimador *non-negative garrote* inclui um fator não-negativo no modelo estimado via MQ. Sua desvantagem é que nas situações em que o estimador MQ não se comporta muito bem, acontece o mesmo com seus resultados, fato que não ocorre com o estimador LASSO.

Diante desta discussão, é possível perceber a importância que o parâmetro de penalização apresenta para o LASSO, e como esta penalização não é previamente conhecida , este fator passa a ser mais uma medida a ser estimada. No artigo original, Tibshirani (1996) apresenta três possibilidades de estimação para s: validação cruzada, validação cruzada generalizada e estimativa de risco, embora o autor afirme que, na prática, não há muita diferença entre as três e que pode ser escolhida a técnica mais conveniente. De qualquer maneira, independente do método escolhido, encontrar um valor ótimo para o parâmetro de penalização é, indiscutivelmente, um ponto fundamental para o bom desempenho da técnica.

Hastie et al. (2015) afirmam que "o limite s na técnica LASSO controla a complexidade do modelo" [2] (tradução nossa), ou seja, variações neste novo parâmetro influenciam diretamente na estimação dos demais. Valores muito altos de s "liberam" mais parâmetros, ou seja, fazem com que uma menor quantidade de coeficientes do modelo sejam anulados, dificultando a interpretação. Inversamente, baixos valores para s deixam o modelo mais esparso, facilitando a interpretação. Em compensação, no que diz respeito a acurácia, valores pequenos podem não captar características importantes dos dados e

[2] "the bound s in the LASSO criterion controls the complexity of the model"

valores altos podem causar sobreajuste (do inglês, *overfitting*), o que, a grosso modo, significa que o modelo não tem um bom poder de generalização. Em todos os cenários extremos, o erro de predição será inflacionado, assim os autores afirmam que "geralmente existe um valor intermediário de s que atinge um bom equilíbrio entre esses dois extremos, e produz um modelo com alguns coeficientes iguais a zero"[3] (tradução nossa). Portanto, a estimação eficiente de s é um aspecto muito importante do LASSO, garantindo melhoras consideráveis no ajuste do modelo.

A seguir faremos uma breve recapitulação sobre o método de mínimos quadrados, afinal, por conta de suas vantajosas propriedades, mesmo que seu uso não seja recomendado, seus estimadores auxiliam como ponto de partida na busca de estimadores melhores. Um bom exemplo disso é justamente a técnica LASSO, pois, embora se coloque como uma alternativa a esses estimadores, parte exatamente deles e busca melhorá-los.

[3] "there is usually an intermediate value of s that strikes a good balance between these two extreme, and in the process, produces a model with some coefficients equal to zero"

3 REGRESSÃO E O MÉTODO DE MÍNIMOS QUADRADOS

Quando pensamos em estudar a relação entre variáveis e prever ou entender seus comportamentos é comum a construção de modelos de regressão. Ajustar um modelo basicamente significa buscar encontrar uma expressão que traduza algebricamente esta possível associação entre as variáveis, o que Friedman et al. (2008) descreve como uma "tarefa de aproximação de funções". O interesse em identificar esta relação está presente no cotidiano de muitos cientistas e pesquisadores.

Vários são os propósitos envolvidos na construção de modelos de regressão, destacando-se: descrição do banco de dados, estimação de parâmetros de interesse, previsões e estimativas e controle. (MONTGOMERY et al., 2012)

Os tipos de função que podem ser utilizados para descrever essa relação são diversos, destacaremos os Modelos de Regressão Linear Simples (MRLS) e os Modelos de Regressão Linear Múltiplos (MRLM), e nas seções a seguir discutiremos sua importância, bem como o método mais comum para a estimação de seus parâmetros, o Método de Mínimos Quadrados (MMQ).

3.1 Modelos de regressão linear

O MRLS estabelece uma relação linear entre os coeficientes de uma função, assumindo a seguinte forma funcional,

$$y_i = \beta_0 + \beta_1 x_i + e_i, \quad \text{com} \quad i = 1, 2, ..., n, \tag{1}$$

em que y_i é a variável resposta dada em função de x_i, chamada de variável preditora. As constantes β_0 e β_1, denominadas coeficientes de regressão, são desconhecidas e a estimação de seus valores constitui um dos problemas centrais da construção do modelo. Já $e_1, \cdots, e_n$ representam os erros aleatórios, os quais se supõe constituir uma sequência de variáveis aleatórias de média zero, variância constante e não correlacionadas.

É importante ressaltar que os modelos são ditos lineares quando a relação estabelecida entre seus coeficentes é linear, o que não significa que a relação entre x e y também o seja. O uso do MRLS apresenta várias vantagens: é simples, fácil de ajustar, é capaz de descrever uma gama de situações e em alguns casos, mesmo que a relação não seja linear, é possível aplicar transformações e linearizar a função.

Caso tenhamos interesse em estudar a relação entre a variável resposta e múltiplas variáveis preditoras, também chamadas de covariáveis, podemos utilizar o MRLM, que de forma análoga ao MRLS, possui forma funcional dada pela igualdade em (2).

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + e_i, \quad \text{com} \quad i = 1, 2, ..., n \tag{2}$$

em que, y_i é a variável resposta, e cada vetor $\mathbf{x}_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{ip})^\top$ representa as p características observadas para cada indivíduo i de nossa amostra. O intercepto, β_0, e o vetor dos coeficientes de regressão, $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_p)^\top$, são os parâmetros a serem estimados, uma vez que, conforme já discutimos, estabelecem a relação que se dá entre a variável resposta e as preditoras.

De maneira geral, a partir de agora, trataremos todos os casos apenas como regressões lineares múltiplas, uma vez que o MRLS nada mais é do que uma regressão linear múltipla com $p = 1$. Com o aumento na quantidade de covariáveis a serem analisadas, é conveniente utilizar a notação matricial, pois, quanto mais aumentamos os valores de n e p, mais difícil se tornará operar com a expressão em (2). Matricialmente, representaremos regressões lineares da seguinte forma:

$$\mathbf{y} = \mathbf{1}_n\beta_0 + \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \tag{3}$$

em que, $\mathbf{y}$ é um vetor $n \times 1$ de variáveis resposta, $\mathbf{1}_n$ representando um vetor de uns, $\mathbf{X}_{n\times p}$ é a matriz de especificação, na qual as linhas representam os indivíduos e cada coluna é uma covariável, β_0 e $\boldsymbol{\beta}$ representam os vetores de parâmetros, finalizando, temos $\mathbf{e}$ que é um vetor $n \times 1$ de erros de medida. Para que a construção do MRLS seja de fato possível é preciso que esses resíduos sejam não correlacionadas, tenham média zero e a variância, denotada por σ^2, deve ser constante. Esta última característica é chamada de homoscedasticidade. (MONTGOMERY, et al. 2012)

Muitas são as técnicas que podem ser aplicadas no ajuste destes modelos, a mais utilizada é o método de mínimos quadrados (MMQ), por apresentar ótimas propriedades. Na seção seguinte, utilizando a notação matricial, discutiremos a estimação dos parâmetros do modelo via MMQ, bem como suas potencialidades e limitações.

3.2 Método de mínimos quadrados

O método de mínimos quadrados consiste em minimizar a soma dos quadrados dos resíduos do modelo, ou seja, $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, e a partir daí estimar β_0 e $\boldsymbol{\beta}$. Geometricamente, o processo é equivalente a minimizar a soma dos quadrados das distâncias dos valores observados à reta que representa o gráfico do modelo ajustado.

Algebricamente, podemos buscar esta otimização a partir da expressão dada em 3, porém, para reduzir os cálculos necessários e facilitar as representações, é conveniente incorporar o intercepto β_0 ao vetor β. Assim, o modelo como qual trabalharemos a partir de então, passa a ser o expresso em (4).

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \tag{4}$$

em que os vetores $\mathbf{y}$ e $\mathbf{e}$ não se alteram e a matriz de especificação $\mathbf{X}$ passa a ter a primeira

coluna formada apenas por uns,

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & \cdots & \cdots & & \cdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \text{ e } \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}.$$

Segundo Montgomery et al.(2012), considerando $n > p$ e assumindo que os resíduos não são correlacionados, $\mathbb{E}(e_i) = 0$ e $\text{Var}(e_i)=\sigma^2$, nosso objetivo é minimizar a expressão em (5):

$$S(\beta_0, \beta_1, ..., \beta_p) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2. \tag{5}$$

A função S é denominada como função de mínimos quadrados e para minimizá-la devemos derivá-la com relação a $\boldsymbol{\beta}$. Em seguida, igualamos essas derivadas a zero, encontrando assim a equação normal de mínimos quadrados, expressa em 6 .

$$\frac{\partial S}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X}\hat{\boldsymbol{\beta}} = 0. \tag{6}$$

A solução da equação em (6) é o vetor de parâmetros estimados via mínimos quadrados, dado pela expressão em (7),

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{7}$$

Mongomery et al. (2012) destacam ainda que é conveniente realizar manipulações algébricas de modo que a matriz $\mathbf{X}$ e o vetor de variável resposta sejam centralizados em relação a média, com variância unitária. Com isso, a matriz $\mathbf{X}^\top \mathbf{X}$, de ordem $p \times p$ e simétrica, passa a ser a matriz de correlações, pois seus elementos que estão fora da diagonal principal representam o coeficiente de correlação entre os pares de variáveis explicativas. Já $\mathbf{X}^\top \mathbf{y}$ representa o vetor de correlações entre as variáveis explicativas e as variáveis resposta.

É importante destacar também que é interessante que a matriz $\mathbf{X}^\top \mathbf{X}$ seja singular, uma vez que para a estimação dos coeficientes de regressão usaremos sua inversa. Nos casos em que essa característica não é observada, faz-se necessário o uso de inversas generalizadas.

Como já foi dito, o estimador de MQ possui propriedades ótimas, destacando-se o fato de ser o melhor estimador linear não viciado(BLUE)[4]. Porém, estes estimadores podem não ser boas opções em algumas situações. Os dois problemas mais comuns que dificultam o uso dos estimadores de MQ são: alta dimensionalidade ($p \gg n$) e covariáveis

[4]Best Linear Unbiased Estimator

muito correlacionadas (multicolinearidade).

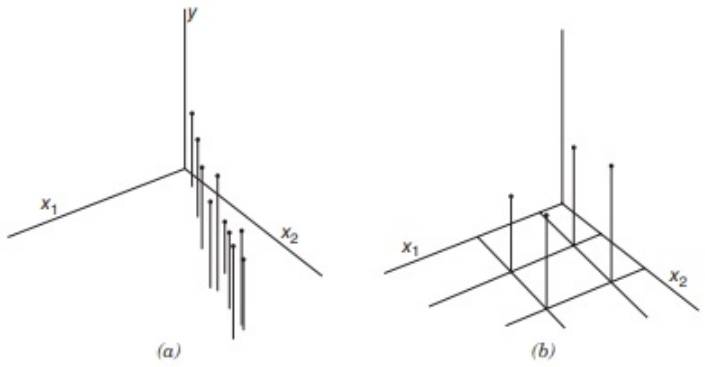
Segundo Bühlmann e Van de Geer (2011), além de a alta dimensionalidade no banco de dados afetar a singularidade da matriz $\mathbf{X}^\top\mathbf{X}$, outro grande problema que essa característica causa no processo de estimação é o fato de não ser possível garantir a unicidade dos estimadores de MQ, o que pode causar sobreajuste.

Já quanto a multicolinearidade, considera-se que o ajuste apresenta esse problema quando há uma relação quase linear entre duas ou mais variáveis explicativas, ou seja, quando as colunas da matriz $\mathbf{X}$ são linearmente dependentes. Montgomery *et al.* (2012) colocam que a "multicolinearidade é uma forma de mau condicionamento da matriz $\mathbf{X}^\top\mathbf{X}$"[5] (tradução nossa). Neste caso, assim como no problema de alta dimensionalidade, a singularidade de $\mathbf{X}^\top\mathbf{X}$ impossibilita a obtenção da inversa tradicional, $(\mathbf{X}^\top\mathbf{X})^{-1}$.

Além disso, a multicolinearidade também afetar a variância de $\hat{\boldsymbol{\beta}}$, tornando-a muito alta (PEREIRA, 2017). A busca pela menor variância é praxe em técnicas de estimação estatísticas, uma vez que altos valores para esta medida significam que diferentes amostras coletadas resultariam em estimativas instáveis para os parâmetros do modelo. Esse problema pode ser ilustrado pela Figura 2, apresentada em Montgomery et al. (2012), na qual exemplifica-se o ajuste de um modelo (x_1, x_2, y), em que x_1 e x_2 são as variáveis explicativas e y a variável resposta . Na Figura 3(a) há multicolinearidade entre as referidas variáveis, já na Figura 3(b) é possível perceber que x_1 e x_2 são ortogonais, uma característica da ausência de multicolinearidade.

Se traçarmos um plano contendo os pontos expressos na Figura 3 (a) é possível perceber que este apresenta-se instável e sensível a mudanças nos dados, mesmo que pequenas, além disso o modelo não é confiável para extrapolações. Em contrapartida, na Figura 3(b) a ortogonalidade entre as variáveis forma um plano de mínimos quadrados bem definido, mais estável. Nesse caso a estimação dos parâmetros se dá de maneira bem definida, sendo β_0 a interseção ente o plano e o eixo y e β as inclinações nas direções de x_1 e x_2.

Figura 2 – Exemplo de ajustes com e sem multicolinearidade, respectivamente.



Fonte: Montgomery et al. (2012)

[5] "multicollinearity is a form of ill - conditioning in the $\mathbf{X}^\top\mathbf{X}$ matrix"

Montgomery et al. (2012) destacam também que a multicolinearidade pode causar "confusões" nos sinais dos coeficientes de regressão, ou seja, características sobre as quais o pesquisador já tenha um conhecimento empírico de que afetam positivamente a variável resposta podem vir a apresentar sinal negativo, e vice versa. Já Casagrande (2016) destaca que a multicolinearidade também pode interferir no teste de Análise de Variância (ANOVA), importante ferramenta estatística para identificar quais as variáveis são significativas no ajuste do modelo. Porém o problema da multicolinearidade pode influenciar o teste a apontar como não significativas variáveis sabidamente importantes.

Os fatores que causam multicolinearidade são diversos e podem ou não ser controlados pelo analista, são exemplos: alta dimensionalidade, a escolha de técnicas de amostragem ou de modelos de regressão não apropriados para a situação a ser estudada, restrições na população ou no modelo. Como já comentamos, todas essas situações dificultam ou impossibilitam o uso dos estimadores de MQ, justificando a busca por métodos alternativos para a estimação dos coeficientes do modelo a ser ajustado.

Um trabalho interessante de comparação entre o uso de várias técnicas para ajustar modelos de regressão com problemas de alta dimensionalidade e/ou multicolinearidade é o estudo de Casagrande (2016). Comparando métodos de seleção (componentes principais e mínimos quadrados parciais) e métodos de encolhimento (regressão em crista e LASSO), o autor analisa fatores como a mediana das médias dos desvios ao quadrado, proporção de vezes em que a técnica obteve a menor distância relativa, quantidade de vezes que cada coeficiente foi considerado zero (apenas para o LASSO) e o tempo computacional, em dados simulados e não-simulados. Para os dados simulados o autor discorre sobre diversos cenários, combinando diferentes intensidades e quantidades de efeitos. A conclusão mais significativa para a técnica que estamos estudando foi que, para dados simulados, o LASSO apresenta um alto índice de acertividade ao zerar covariáveis que de fato apresentavam efeito nulo, em todos os cenários estudados, destacando o caráter de seleção de covariáveis que a técnica apresenta, embora este índice diminua a medida que a quantidade de efeitos não-nulos aumenta.

Nos próximos capítulos discutiremos algumas possibilidades de penalizações que podem melhorar a estimação, uma vez que o MMQ não se mostre adequado. A partir de explorações algébricas e geométricas, trataremos algumas vantagens e desvantagens das diferentes formas de penalização de um modelo, como podem ser aplicadas a fim de melhorar o desempenho dos estimadores de MQ, além de buscar justificar as vantagens que penalizações do tipo L_1 podem trazer para o ajuste. Posteriormente daremos maior ênfase a discussão da técnica LASSO, que é o objetivo principal deste trabalho.

4 PENALIZAÇÕES L_q

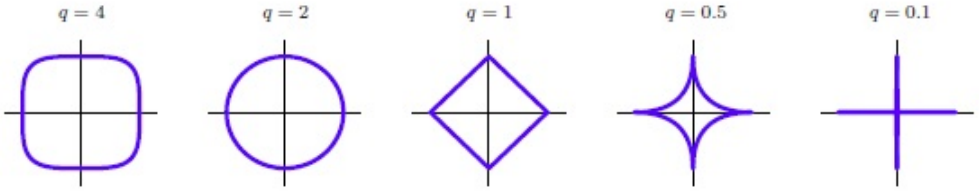
Sob as mesmas suposições do MRLM, o método de mínimos quadrados penalizados apresenta-se como uma possibilidade de refinamento do MMQ, conforme viemos discutindo desde o ínicio deste trabalho. Essa penalização pode se dar conforme o critério a seguir:

$$\min_\beta \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + s \sum_{j=1}^{p} |\beta_j|^q \right\}. \tag{8}$$

Perceba que o termo $s \sum_{j=1}^{p} |\beta_j|^q$ em (8), representa a penalização sugerida. Frank e Friedman[6](1993 *apud* TIBSHIRANI, 1996) desenvolveram um estudo sobre o comportamento da região de restrição delimitada pelo fator de penalização para diferentes valores de q, e denominaram esta generalização de *Bridge Regression*. Alguns exemplos dos formatos destas regiões podem ser vistos nas Figuras 3 e 4.

O caso $q = 0$ contabiliza a quantidade de coeficientes não-nulos presentes no modelo, e equivale ao método *best-subset selection*, podendo também ser considerado como uma penalização do tipo L_0. Para $q = 1$ e $q = 2$ temos, respectivamente, penalizações do tipo L_1 e L_2, que correspondem respectivamente ao LASSO e a regressão em cristas. Neste contexto, o LASSO torna-se especial por ser o menor valor de q, que mais se aproxima do *best-subset selection*, com a propriedade de conduzir a uma região de restrição convexa e, consequentemente, a um problema de otimização convexo. No geral, para $q < 1$, as regiões de restrição serão não convexas, e para $q > 1$ os problemas serão convexos.

Figura 3 – Regiões de restrição para diferentes valores de q em $\mathbb{R}^2$.



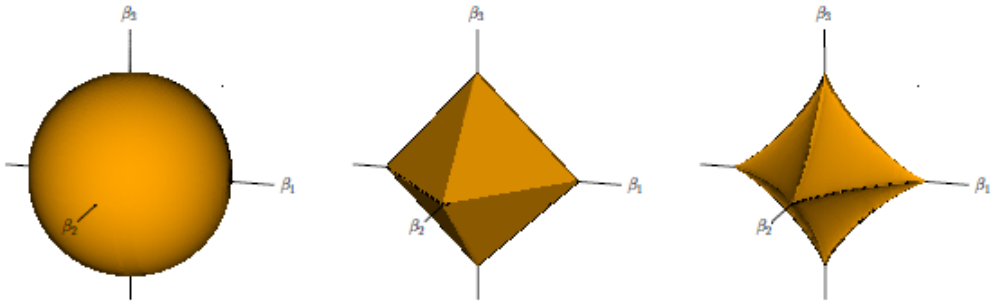
Fonte: Hastie et al. (2015)

Hastie et al. (2015) ressaltam o proveito de trabalhar com funções convexas, uma vez que "esta inequação garante que uma função convexa não possa ter um ponto de mínimo local que não seja ponto de mínimo global"[7] (tradução nossa). Além disso, nos casos em que a função não é diferenciável em todos os pontos, a convexidade garante que seja possível encontrar aproximações para o limite inferior, embora não seja possível

[6]Frank, I. and J. Friedman (1993). A statistical view of some chemometrics regression tools. Technometrics 35(2), 109-135.

[7]"This inequality guarantees that a convex function cannot have any local minima that are not also globally minimal."

Figura 4 – Regiões de restrição em $\mathbb{R}^3$ para valores de q=$\{2, 1$ e $0{,}8\}$, respectivamente.

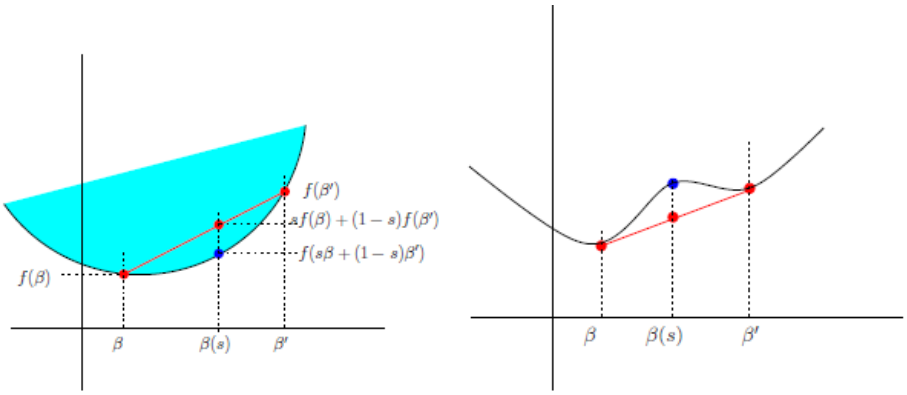


Fonte: Hastie et al. (2015)

calculá-lo efetivamente. Para tanto, usa-se a ideia de subgradientes, a qual discutiremos nos capítulos futuros.

A Figura 5 ilustra, a partir de uma visualização geométrica, a diferença de otimização de uma função convexa e uma não convexa.

Figura 5 – Exemplo gráfico de uma função convexa e uma função não convexa.



Fonte: Hastie et al. (2015)

Ainda sobre as vantagens de manipular regiões convexas, os autores afirmam que "a convexidade simplifica muito a computação tanto quanto a esparsidade. Isto permite que algoritmos escaláveis resolvam até mesmo problemas com milhões de parâmetros"[8] (tradução nossa).

Finalizamos esta discussão ressaltando que penalizações não-convexas podem ser oportunas quando p for muito grande e a quantidade de covariáveis significativas for pequena. Neste caso, na tentativa de alcançar e restringir todas as variáveis não significativas, o lasso pode acabar restringindo demais até mesmo as que são relevantes. Neste contexto, uma saída natural seria recorrer às penalizações L_q, com $0 < q < 1$, mesmo que apresentem resoluções trabalhosas.

Hastie et al. (2015) sugerem duas possibilidades de penalizações não convexas

[8] "Convexity greatly simplifies the computation, as does the sparsity assumption itself. They allow for scalable algorithms that can handle problems with even milions of parameters"

a serem abordadas, o método SCAD (*smootly clipped absolute deviation*) e MC+ (*minimax concave*), sobre os quais não nos aprofundaremos pois fogem do escopo deste trabalho.

4.1 Comparação entre as penalizações L_1 e L_2

Na seção anterior discutimos as vantagens da utilização de penalizações que resultem em problemas de otimização convexos, pois geram problemas computacionalmente mais tratáveis, no sentido de menor tempo e esforço computacional. A seguir, trataremos da característica de produzir modelos esparsos, da qual gozam as penalizações L_1.

Para facilitar o entendimento, faremos uma comparação entre as penalizações do tipo L_1 e L_2, ou seja, LASSO e regressão em crista, através da Figura 6, que, embora aborde apenas o caso bidimensional, é bem elucidativa quanto a uma das principais características do LASSO, que é o encolhimento de coeficientes. Perceba que o critério de minimização dado pela equação (5), sob suposição de normalidade, pode ser expresso na forma matricial a partir da forma quadrática

$$(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)^\top \mathbf{X}^\top \mathbf{X} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0). \tag{9}$$

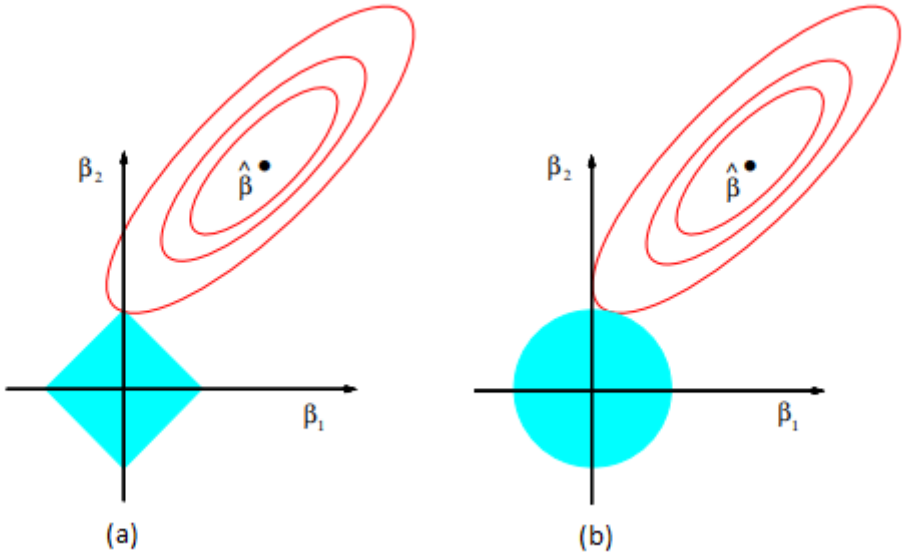
As elipses em vermelho representam os elipsóides de confiança, ou seja, configuram o lugar geométrico em que a soma dos quadrados dos resíduos, expressa pela função em (9), é uma constante diferente do mínimo, este, ocorre no centro desta elipse, que é o estimador de MQ. O quadrado em azul na Figura 6 (a) representa a região de restrição a partir da técnica LASSO, dada pela expressão $|\beta_1| + |\beta_2| \leq s$. Já a Figura 6 (b) representa a região de restrição dada pela regressão em crista, cuja expressão é a equação de uma circunferência de raio s, ou seja, $\beta_1^2 + \beta_2^2 \leq s$.

Independente do formato da região de restrição, Murphy(2012) coloca que "da teoria da otimização restrita, sabemos que a solução ideal ocorre no ponto em que o menor nível definido da função objetivo intersecta a superfície de restrição" [9] (tradução nossa). Em nosso estudo, isso significa que a solução da otimização, ou seja, a minimização da soma dos quadrados dos resíduos, se dá no primeiro momento em que a elipse intercepta a região de restrição, um vez que este ponto produz o vetor de coeficientes de regressão com a menor norma consistente com um aumento específico na soma dos quadrados dos resíduos (MONTGOMERY et al., 2012).

Um ponto fundamental dessa discussão é que, quando essa interseção acontece no vértice o coeficiente é zero. E é aí que o LASSO se destaca, uma vez que os discos produzidos pela região de restrição da regressão em crista não possuem "pontas", logo, embora possam apresentar coeficientes bem pequenos, estes nunca serão zero. Quando

[9] "from the theory of constrained optimization, we know that the optimal solution occurs at the point where the lowest level set of the objective function intersects the constraint surface"

Figura 6 – Comparação entre penalizações L_1 e L_2 em $\mathbb{R}^2$

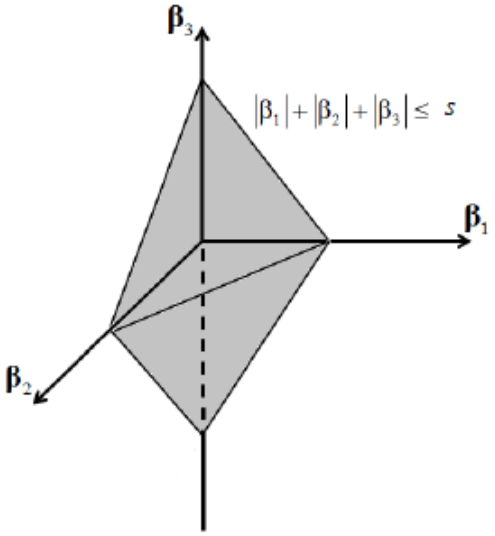


Fonte: Hastie et al. (2015)

$p > 2$, as regiões de restrição do LASSO serão hipercubos, com ainda mais "pontas" e consequentemente maior probabilidade de $\hat{\beta}_j = 0$, ao passo que para a regressão em cristas esta região terá a forma de hiperesferas.

Pereira (2017) desenvolveu um importante estudo sobre abordagens geométricas para alguns métodos de regressão, dentre estes o LASSO. A Figura (7) foi retirada de seu trabalho e mostra o caso em que $p = 3$. Obviamente, quanto mais aumentarmos o valor de p, mais esta visualização espacial fica complexa, e teremos que nos apoiar em suportes algébricos.

Figura 7 – Região de restrição do LASSO em $\mathbb{R}^3$.



Fonte: Pereira (2017).

Portanto, esta importante propriedade de tornar o modelo esparso, consequentemente selecionando variáveis e facilitando a interpretação do modelo, torna o LASSO uma poderosa técnica para ajuste de modelos de regressão, especialmente aqueles que descrevem casos pouco convencionais. Nos próximos capítulos iremos discutir as estimações mais importantes para sua aplicação.

5 ESTIMAÇÕES LASSO EM MODELOS LINEARES.

Tibshirani(1996) propôs buscar estimar o vetor $\boldsymbol{\beta}^\top$ associado ao MRLM dado em (2), através de uma penalização do tipo L_1, com $s \geq 0$, minimizando a expressão:

$$\frac{1}{2N}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2, \quad \text{sujeito a} \sum_{j=1}^{p}|\boldsymbol{\beta_j}| \leq s. \tag{10}$$

A fim de anular o efeito que possíveis diferenças de unidades nas mensurações das covariáveis possam causar no estimador LASSO, assim como fizemos na estimação dos coeficientes de regressão de MQ, é conveniente padronizar as colunas da matriz de especificação, centralizando cada coluna em sua média e garantindo que a variância seja unitária, assim teremos $\frac{1}{N}\sum_{i=1}^{N} x_{ij} = 0$ e $\frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 = 1$. Se aplicarmos a mesma padronização no vetor de variáveis resposta, ou seja $\frac{1}{N}\sum_{i=1}^{N} y_i = 0$, teremos mais uma vantagem, que é a omissão do intercepto do modelo. Este "deslocamento", em nada altera a estimação de $\boldsymbol{\beta}$, e por outro lado, o estimador do intercepto é dado por:

$$\hat{\beta}_0 = \overline{y} - \sum_{j=1}^{p}\overline{x}_j\hat{\boldsymbol{\beta}}_j,$$

em que $\overline{y}$ é a média da variável resposta e $\overline{x}_j$ é a média original de cada x_j, antes da padronização.

Considerando a penalização proposta por Tibshirani(1996) e as padronizações citadas, podemos reexpressar o problema de estimação LASSO da seguinte forma, também chamada Lagrangeana:

$$\frac{1}{2N}\|Y - X\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \quad \text{com} \quad \lambda \geq 0. \tag{11}$$

Perceba que em (11) o fator de penalização passa a ser o parâmetro λ. Hastie et al. (2015) ressaltam que há uma correspondência biunívoca entre os valores de s e λ, ou seja, para cada valor de s que satisfaça a restrição $\|\boldsymbol{\beta}\|_1 \leq s$ existirá um único valor correspondente para λ que produza a mesma solução para a expressão na forma de Lagrange.

Como já foi discutido, o estimador LASSO é o ponto da região de restrição que está a menor distância dos estimadores de MQ, os quais representam o centro de uma elipse.

Discutiremos a estimação de s através do método da validação cruzada, além de analisar que contribuição a discussão sobre os graus de liberdade do modelo pode trazer para a estimação deste parâmetro. Quanto a $\hat{\boldsymbol{\beta}}$ apresentaremos duas possibilidades de abordagem para sua obtenção, uma algébrica, via subgradientes e suavização de limites e a outra geométrica, via interseção de lugares geométricos.

5.1 Estimação de $\boldsymbol{\beta}$

Para estimar os coeficientes do modelo apresentaremos nos tópicos a seguir duas possibilidades de abordagem, uma algébrica, tendo como suporte Murphy(2012) e Hastie et al. (2015) e uma interpretação geométrica dessa estimação, a partir dos estudos de Pereira (2017).

5.1.1 Interpretação algébrica do estimador $\boldsymbol{\beta}$

Recapitulando a discussão, nosso objetivo principal é minimizar a expressão em (11), e sabemos que, se uma função é contínua, diferenciável e com $f^{(1)} \neq 0$ para todos os pontos de seu domínio, sua derivada de primeira ordem é ponto de mínimo local. E mais, se a função é convexa, o ponto de mínimo local também é mínimo global. No caso em que estamos interessados, esse resultado significa que é possível encontrar um $\hat{\boldsymbol{\beta}}$ que minimize (11).

A questão chave desse problema é que a função objetivo nem sempre é diferenciável em $\lambda\|\boldsymbol{\beta}\|_1$. Nestas circunstâncias, não podemos simplesmente utilizar a ideia clássica de derivar a função, igualar a zero e determinar $\hat{\boldsymbol{\beta}}$. Uma solução para esta problemática é estender a ideia de derivada e trabalhar com o conceito de subgradiente.

Inicialmente, vamos definir subgradientes, dada uma função $f : \mathbb{R}^p \longrightarrow \mathbb{R}$ e seja z uma escalar, diremos que $\boldsymbol{\beta}_l$ é um subgradiente de f, sempre que:

$$f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}_l) \geq z(\boldsymbol{\beta} - \boldsymbol{\beta}_l), \qquad \text{em que} \quad \forall \boldsymbol{\beta}_l \in \mathbb{R}^p. \tag{12}$$

Chamamos de subdiferencial (*subdifferential*) o conjunto de todos os subgradientes de f, dado pelo intervalo [a,b], em que:

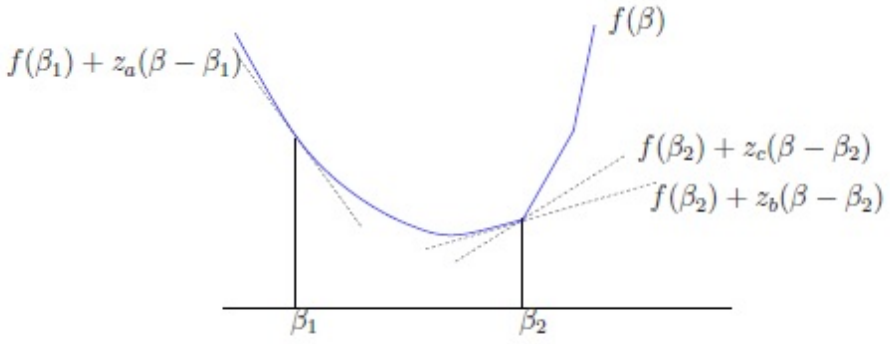
$$a = \lim_{\boldsymbol{\beta} \to \boldsymbol{\beta}_l^-} \frac{f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}_l)}{\boldsymbol{\beta} - \boldsymbol{\beta}_l} \quad \text{e} \quad b = \lim_{\boldsymbol{\beta} \to \boldsymbol{\beta}_l^+} \frac{f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}_l)}{\boldsymbol{\beta} - \boldsymbol{\beta}_l}.$$

Assim, analisamos as possibilidades para $\boldsymbol{\beta}_l$ a fim de determinar o subdiferencial de f, e o denotamos por $\frac{\partial f}{\partial \boldsymbol{\beta}_l}$. Nos casos em que f é diferenciável em todos os seus pontos, o subdiferencial nada mais é do que a derivada parcial de f com relação a $\boldsymbol{\beta}_l$.

Na Figura 8, Hastie et al. (2015) ilustram os subgradientes de uma função qualquer $f : \mathbb{R} \longrightarrow \mathbb{R}$. Perceba que a função é diferenciável em $\boldsymbol{\beta}_1$, logo o subgradiente de f é único, e sua subderivada é simplesmente $f^{(1)}(\boldsymbol{\beta}_1)$. Já em $\boldsymbol{\beta}_2$ a função não é diferenciável, assim existem várias tangentes a f neste ponto, e cada uma delas apresenta um limite inferior. Fazendo uma analogia com o cálculo tradicional, é como se houvesse várias derivadas de f em $\boldsymbol{\beta}_2$, e o conjunto de todos elas é o que estamos denominando subdiferencial de f em $\boldsymbol{\beta}_2$, $\frac{\partial f}{\partial \boldsymbol{\beta}_2}$.

De volta a função objetivo dada em (11), a dualidade de Lagrange garante que

Figura 8 – Ilustração de gradientes e subgradientes.



Fonte: Hastie et al. (2015)

$\boldsymbol{\beta}_l$ será um ponto de mínimo sempre que sua derivada com relação a $\boldsymbol{\beta}_l$ seja zero. Neste caso específico o primeiro termo de (11)é derivável em todos os pontos, já o segundo não é, o que caracteriza a função como não-suave e motiva o uso de subgradientes. (MURPHY, 2012)

Para resolver este problema, o autor divide a solução em duas partes. Calcula a derivada do termo quadrático, incialmente, que é dada por:

$$\frac{\partial \|y - \mathbf{X}\boldsymbol{\beta}\|_2^2}{\partial \boldsymbol{\beta}_j} = a_j \boldsymbol{\beta}_j - c_j \qquad \text{em que:} \tag{13}$$

$$a_j = 2\sum_{i=1}^{n} x_{ij}^2, \tag{14}$$

$$c_j = 2\sum_{i=1}^{n} x_{ij}(y_i - \boldsymbol{\beta}_{-j}^{\top} \cdot x_{i,-j}). \tag{15}$$

em que $\boldsymbol{\beta}_{-j}$ é o vetor de parâmetros $\boldsymbol{\beta}$, sem o coeficiente que corresponde a j-ésima variável explicativa, bem como $\mathbf{x}_{i,-j}$ são os elementos da matriz de especificação, retirando-se a j-ésima coluna.

O termo c_{ij} apresenta um papel muito importante em nossa análise, pois capta a relação de proporcionalidade existente entre a variável j e $\mathbf{r}_{-j}$, em que $\mathbf{r}_{-j} = \mathbf{y} - \mathbf{X}_{-j}\boldsymbol{\beta}_{-j}$ e representa os resíduos do modelo, depois de retirada a variável em questão. A magnitude de c_{ij} representa o quanto a variável j interfere em $\hat{\mathbf{y}}$.

Acrescentando a penalização, e derivando a função objetivo por completo, temos:

$$\frac{\partial f}{\partial \beta_j} = a_j \beta_j - c_j + \lambda \frac{\partial \|\boldsymbol{\beta}\|_1}{\partial \beta_j} \tag{16}$$

$$= \begin{cases} \{a_j\beta_j - c_j - \lambda\}, & \text{se } \beta_j < 0. \\ [-c_j - \lambda, -c_j + \lambda], & \text{se } \beta_j = 0. \\ \{a_j\beta_j - c_j + \lambda\}, & \text{se } \beta_j > 0. \end{cases} \tag{17}$$

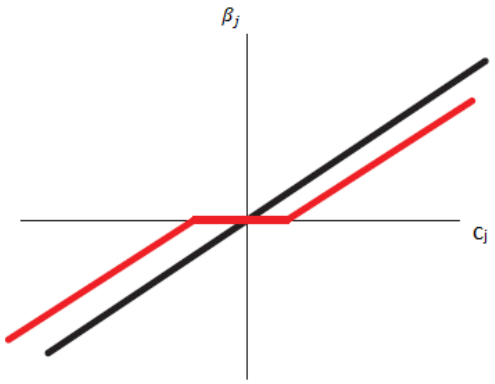
Cada uma das equações de (17) representa um subgradiente da expressão $\frac{\partial \|\boldsymbol{\beta}\|_1}{\partial \beta_j}$, a depender do valor que β_j assume. Igualando cada uma delas a zero, conforme a dualidade de Lagrange estabelece, temos três possibilidades de estimadores para $\boldsymbol{\beta}$, em função de c_j:

$$\hat{\beta}_j = \begin{cases} \frac{c_j+\lambda}{a_j}, & \text{se } c_j < -\lambda. \\ 0, & \text{se } c_j \in [-\lambda, \lambda]. \\ \frac{c_j-\lambda}{a_j}, & \text{se } c_j > \lambda. \end{cases} \tag{18}$$

Se $c_j < -\lambda$, constatamos uma forte correlação negativa entre a j-ésima característica e os resíduos, analogamente, $c_j > \lambda$ indica uma forte correlação positiva. O caso em que $c_j = \lambda$ indica que não há correlação entre as medidas estudadas.

A relação entre c_j e $\hat{\beta}_j$ pode ser visualizada graficamente na Figura (9), com c_j no eixo das abcissas e $\hat{\beta}_j$ no eixo das ordenadas. Perceba que em (16), quando $\lambda = 0$ temos o estimador de MQ, o qual denotaremos $\hat{\beta}^0$, e neste caso seria igual a $\frac{c_j}{a_j}$. Assim, a equação da reta expressa na cor preta é dada por $\hat{\beta}_j^{\,0} = \frac{c_j}{a_j}$, ou seja, corresponde aos estimadores de mínimos quadrados. Já a função cujo gráfico está expresso em vermelho determina a variação dos estimadores LASSO a medida que c_j varia. Podemos perceber que para valores de $c_j \in [-\lambda, \lambda]$, os respectivos $\hat{\beta}_j$ são "encolhidos" a zero, e consequentemente, podemos considerar que não apresentam significância no ajuste do modelo.

Figura 9 – Ilustração referente a suavização de limites.



Fonte: Murphy (2012)

O estimador $\hat{\beta}$ pode ser reescrito através da expressão $\hat{\beta}_j = \text{soft}(\frac{c_j}{a_j})$, em que a função soft() representa o operador de suavização de limites (*soft thresholding*) e é dado

por:

$$\begin{aligned}\hat{\beta}_j &= \text{sign}\left(\frac{c_j}{a_j}\right)\max\left(0, |\frac{c_j}{a_j}| - \frac{\lambda}{a_j}\right) \\ &= \text{sign}\left(\frac{c_j}{a_j}\right)\left(|\frac{c_j}{a_j}| - \frac{\lambda}{a_j}\right)^+. \end{aligned} \tag{19}$$

Considerando que padronizamos os dados, da equação 14 temos que $a_j = 2$, e lembrando que $\hat{\beta_j}^0 = \frac{c_j}{a_j}$, finalmente chegamos a uma expressão para os estimadores LASSO:

$$\hat{\beta}_j = \text{sign}\left(\hat{\beta}^0\right) \cdot \left(|\hat{\beta}^0| - \frac{\lambda}{2}\right)^+. \tag{20}$$

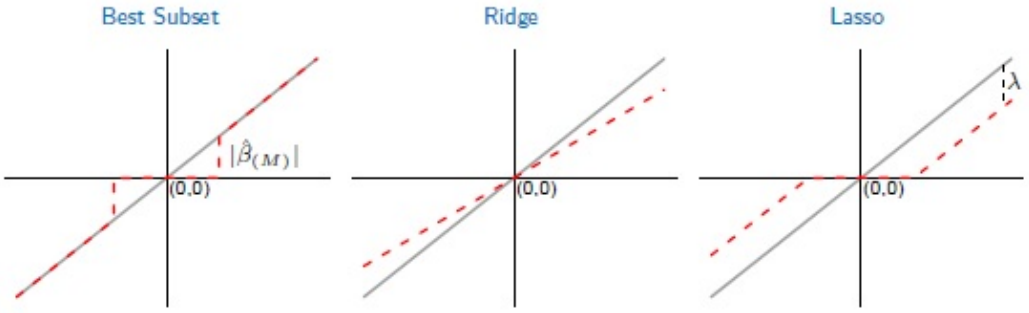
Hastie et al. (2015) observam que para o caso em que $\mathbf{X}$ é uma matriz ortogonal, ou seja $\mathbf{X}^\top\mathbf{X} = \mathbf{I}_n$, em que $\mathbf{I}_n$ representa a matriz identidade de ordem n, o estimador LASSO possui um solução analítica e não necessita de iterações para sua estimação, além disso, quando aplicamos a suavização de limites em matrizes ortogonais o caso multivariado comporta-se como univariado.

No casos em que a matriz $\mathbf{X}$ é não-ortogonal serão necessárias algumas iterações, atualmente facilitadas por algoritmos computacionais. Os autores sugerem diversas classes de procedimentos. Citaremos duas delas por serem amplamente utilizadas nas literaturas mais recentes, além de serem as técnicas utilizadas nos pacotes do `R` que foram implementados para trabalhar com o LASSO, os quais discutiremos nos próximos capítulos, são elas: *coordinate descent* e *homotopy methods*, este último com destaque para o algoritmo *Least Angle Regression*(LARS).

Finalizamos as discussões deste tópico ressaltando a importância que o operador de suavização de limites tem para a técnica que estamos discutindo. Hastie et al. (2015) pontuam que "o operador de suavização de limites desempenha um papel central no LASSO." [10](tradução nossa). Essa importância fica ainda mais clara quando comparamos o papel do parâmetro de ajuste nas penalizações L_0, L_1 e L_2, referentes ao *best-subset selection*, LASSO e regressão em cristas, respectivamente, na estimação de $\hat{\beta}$, através da Figura 10. É possível perceber a capacidade do estimador LASSO de selecionar variáveis, quando "força" alguns coeficientes a zerar através da técnica citada. A regressão em cristas não apresenta essa característica, e o *best-subset*, como já comentamos, não gera uma região de restrição convexa.

[10] "the soft-thresholding operator plays a role central in the LASSO"

Figura 10 – Suavização de limites: comparação entre best-subset selection, regressão ridge e lasso.



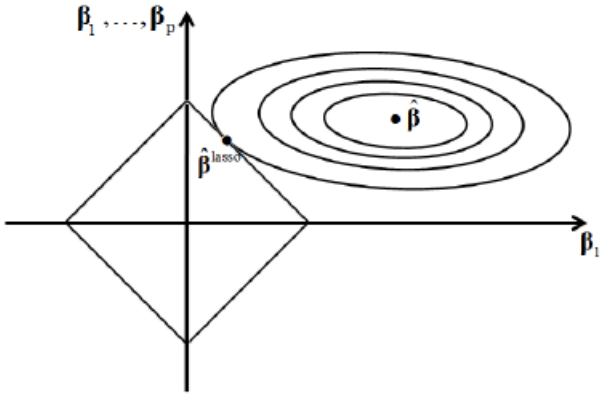
Fonte: Friedman et al. (2008).

5.1.2 Interpretação geométrica do estimador $\boldsymbol{\beta}$

Uma análise geométrica dos estimadores LASSO pode auxiliar bastante na compreensão da técnica. Neste sentido, Pereira (2017) desenvolveu um importante estudo sobre a geometria de alguns métodos de regressão, dentre estes o LASSO, principal objeto de nosso estudo e no qual focaremos nossa atenção.

Geometricamente, buscar o estimador LASSO, significa encontrar o ponto que apresenta menor distância aos estimadores de MQ, os quais denotaremos por $\hat{\boldsymbol{\beta}}^0$, e é interseção entre a região de restrição e uma elipse centrada nestes estimadores, conforme representado na Figura 11.

Figura 11 – Representação geométrica da obtenção do estimador LASSO.



Fonte: Pereira (2017).

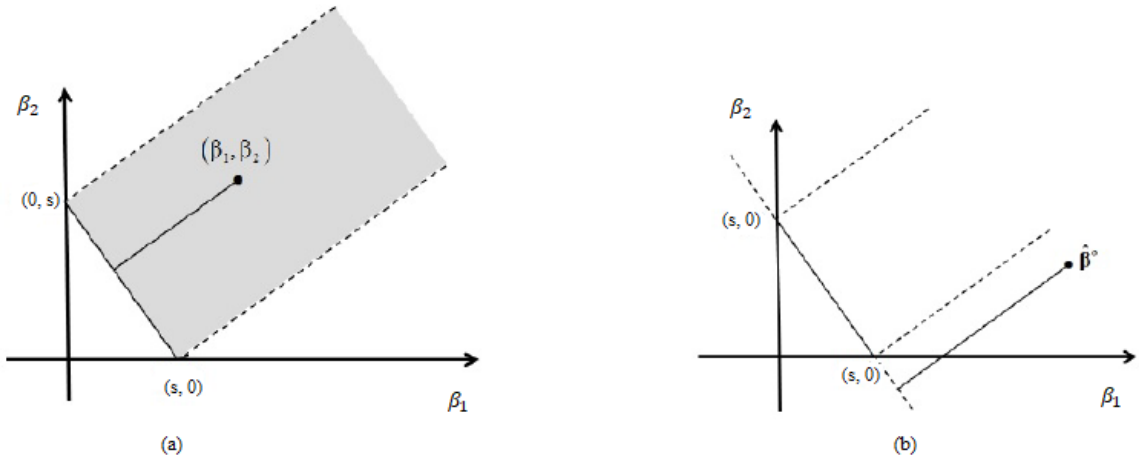
A distância entre $\hat{\boldsymbol{\beta}}$ e $\hat{\boldsymbol{\beta}}^0$ é dada pela métrica $< \mathbf{u}_1, \mathbf{u}_2 >_p = \mathbf{u}_1^\top \mathbf{X}^\top \mathbf{X} \mathbf{u}_2$. Nos casos em que $\mathbf{X}$ é ortogonal, essa distância trata-se do produto interno usual, facilitando os cálculos e interpretações.

Neste tópico analisaremos a estimação de $\boldsymbol{\beta}$ nos casos bidimensionais, em que $\mathbf{X}$ apresenta ou não ortogonalidade. Trabalharemos com $p = 2$ pois, quanto mais aumentamos as dimensões da análise, mais difícil se torna dar uma interpretação geométrica.

Discussões envolvendo situações com dimensões maiores podem ser encontradas em Pereira (2017), por exemplo.

Inicialmente consideraremos que $\mathbf{X}$ é ortogonal, geometricamente isso significa que o estimador de MQ projeta-se perpendicularmente na região de restrição. Neste caso, pela Figura 12, nota-se duas situações possíveis para $\hat{\boldsymbol{\beta}}^0$, ou este pertencerá a região hachurada ou não. A análise será feita considerando que $\hat{\boldsymbol{\beta}}^0$ é um ponto localizado no primeiro quadrante, os demais casos são análogos.

Figura 12 – Projeções do estimador de MQ sobre a região de restrição no caso ortogonal.



Fonte: Pereira, (2017)

Quando os estimadores de MQ encontram-se no retângulo hachurado, conforme ilustrado na Figura 12 (a), suas projeções se dão na aresta do quadrado que representa a região de restrição, cuja equação é dada por $r : \beta_1 + \beta_2 = s$. Considerando a reta w, que passa por $\hat{\boldsymbol{\beta}}^0$ e é perpendicular a r, temos que seus pontos possuem coordenadas do tipo $(\hat{\boldsymbol{\beta}}^0{}_1 + w', \hat{\boldsymbol{\beta}}^0{}_2 + w')$, sendo w' uma escalar e $w' \leq 0$, admitindo o valor zero exatamente no ponto que representa os estimadores de MQ. Conforme mostrado na Figura 11, o estimador LASSO é dado pela interseção entre r e w, para encontrá-lo temos:

$$\begin{aligned} \hat{\beta}_1^0 + w' + \hat{\beta}_2^0 + w' &= s \\ 2w' &= s - (\hat{\beta}_1^0 + \hat{\beta}_2^0) \\ w' &= \frac{s - (\hat{\beta}_1^0 + \hat{\beta}_2^0)}{2}. \end{aligned} \tag{21}$$

Substituindo o valor de w' encontrado em (21), temos que o estimador LASSO é dado pela expressão:

$$\begin{aligned}\hat{\beta} &= \Big(\hat{\beta}_1^0 + \frac{(s - \hat{\beta}_1^0 + \hat{\beta}_2^0)}{2}, \hat{\beta}_2^0 + \frac{(s - \hat{\beta}_1^0 + \hat{\beta}_2^0)}{2}\Big) \\ &= \Big(\frac{s}{2} + \frac{(\hat{\beta}_1^0 - \hat{\beta}_2^0)}{2}, \frac{s}{2} - \frac{(\hat{\beta}_1^0 - \hat{\beta}_2^0)}{2}\Big). \end{aligned} \tag{22}$$

Já na Figura 12 (b), percebemos que quando $\hat{\boldsymbol{\beta}}^0$ está fora da área hachurada, a reta que o contém e é perpendicular a r, intersecta r apenas no segundo ou no quarto quadrante, neste caso é possível perceber que a menor distância se dá ou em (s,0) ou em (0,s), situação na qual $\hat{\beta}_1$ ou $\hat{\beta}_2$ serão zero, respectivamente. É importante ressaltar, mais uma vez, que nesta situação está caracterizada a capacidade de "zerar" coeficientes que o LASSO tem, esse fato ocorre nos vértices da região de restrição, aumentando a dimensão teremos mais vértices, ou seja, mais possibilidades de ter algum coeficiente nulo.

Unindo os dois casos em uma única expressão para o estimador LASSO, temos:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \left(\max\Big(0, \frac{s}{2} + \frac{(\hat{\beta}_1^0 - \hat{\beta}_2^0)}{2}\Big), \max\Big(0, \frac{s}{2} - \frac{(\hat{\beta}_1^0 - \hat{\beta}_2^0)}{2}\Big)\right) \\ &= \left(\Big(\frac{s}{2} + \frac{(\hat{\beta}_1^0 - \hat{\beta}_2^0)}{2}\Big)^+, \Big(\frac{s}{2} - \frac{(\hat{\beta}_1^0 - \hat{\beta}_2^0)}{2}\Big)^+\right). \end{aligned} \tag{23}$$

O caso em que $\hat{\boldsymbol{\beta}}^0$ não é ortogonal à região de restrição está exemplificado na Figura 13. Pereira (2017) sugere que o estimador LASSO será dado da mesma maneira que no caso ortogonal, ou seja, buscando a interseção entre a aresta do quadrado e uma reta que contém $\hat{\boldsymbol{\beta}}^0$. A diferença é que agora, a equação desta reta será dada por $\hat{\boldsymbol{\beta}}^0 + w''(1, a)$, em que o vetor (1,a) é genérico, determinado pela expressão:

$$(-1, 1)(\mathbf{X}^\top \mathbf{X}) \begin{pmatrix} 1 \\ a \end{pmatrix} = 0. \tag{24}$$

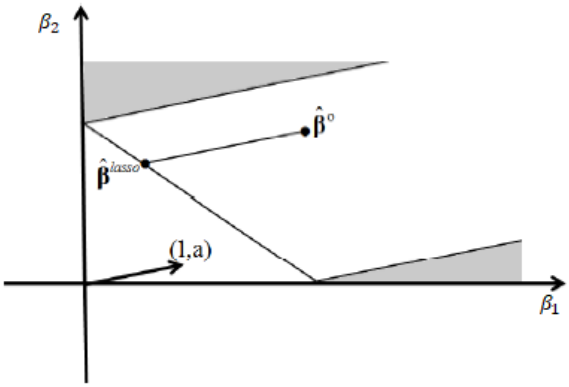
Uma vez determinado o valor de a, encontra-se o valor de w'' e de s, para em seguida determinar $\boldsymbol{\beta}$, analogamente ao caso ortogonal.

5.2 Estimação de s

O parâmetro de ajuste s desempenha um papel fundamental na técnica LASSO, pois "controla a quantidade de encolhimento aplicada às estimativas"[11] (TIBSHIRANI, 1996), (tradução nossa). Quanto maior for seu valor, menor será a distância entre $\hat{\boldsymbol{\beta}}^0$ e $\hat{\boldsymbol{\beta}}$, fazendo com que os estimadores LASSO tendam a se aproximar, ou até mesmo atingir, os estimadores de MQ, esses inclusive representam o maior valor que aqueles po-

[11] "[the parameter $s \geq 0$] controls the amount of shrinkage that is applied to the estimates"

Figura 13 – Representação de $\hat{\boldsymbol{\beta}}^0$ não ortogonal à região de restrição.



Fonte: Pereira, (2017)

dem assumir, ou seja, a medida que s aumenta, $\hat{\boldsymbol{\beta}}$ se aproxima de $\hat{\boldsymbol{\beta}}^0$. Em contrapartida, quanto menores forem os valores de s, menor será a área de restrição, diminuindo também a probabilidade de $\hat{\boldsymbol{\beta}}$ alcançar $\hat{\boldsymbol{\beta}}^0$. Portanto, quanto mais s se aproximar de zero, mais esparso o modelo ficará.

Considerando o modelo na forma Lagrangeana apresentada em (11), devemos ter um pouco de cuidado com a interpretação do parâmetro de ajuste, agora denotado por λ, pois esta será diferente. Neste caso, quando λ for igual a zero, o termo $\lambda\|\boldsymbol{\beta}\|_1$ irá zerar, e $\hat{\boldsymbol{\beta}}$ será exatamente igual a $\hat{\boldsymbol{\beta}}^0$. A medida que os valores de λ aumentem, o modelo vai ficando esparso e tendendo ao modelo nulo.

Desde a primeira proposta do autor, várias técnicas já foram apresentadas para a estimação deste parâmetro. Neste trabalho abordaremos duas possibilidades: a validação cruzada (*cross-validation*) e os graus de liberdade do LASSO.

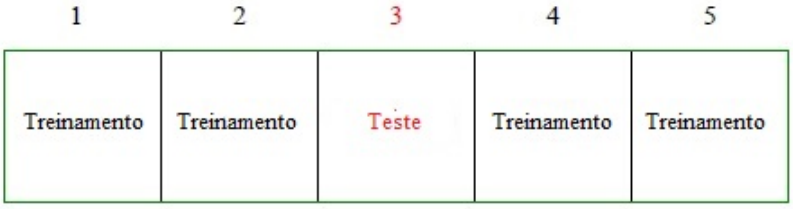
5.2.1 Estimação de s via validação cruzada

O método denominado validação cruzada (*cross-validation*) baseia-se na divisão aleatória do vetor de respostas em k grupos. Essa quantidade de grupos pode variar de 1 a n, e quanto maior for este valor melhor será a precisão da estimação, na literatura comumente sugere-se k igual a 5, 10 ou n. (FRIEDMAN, 2008)

A técnica consiste em, a cada rodada, separar um grupo para ser o grupo de teste (*test set*) e os demais $k-1$ comporão o grupo de treinamento (*training set*), de modo que esse procedimento se repita k vezes e todos os grupos sejam o grupo de teste uma vez. A Figura 14 apresenta uma ilustração da validação cruzada com 5 grupos.

Para cada rodada da validação cruzada o grupo de treinamento será usado para estimar os coeficientes LASSO e ajustar modelos para uma gama de valores de s. Em seguida, aplica-se os valores do grupo de teste nesta gama de modelos e compara-se os valores reais do grupo com os valores preditos para encontrar o erro de predição e o

Figura 14 – Exemplificação da validação cruzada com 5 grupos.



Fonte: Friedman et al. (2008).

erro quadrático médio. Os valores de s aplicados são arbitrários e o ideal é que sejam muitos.

A fim de facilitar os cálculos e a interpretação, Tibshirani(1996) propõe utilizar um parâmetro padronizado s', dado por $s' = \frac{s}{\sum \hat{\beta}_j}$. O ganho de se trabalhar com s' ao invés de s é retirar o peso que diferenças de magnitude entre as variáveis podem apresentar, pois $s \in [0, \hat{\boldsymbol{\beta}}^0]$, enquanto $s' \in [0, 1]$. Já Hastie et al. (2015) utilizam um valor relativo de s, o qual os autores denotaram por $\tilde{s}$, dado por $\tilde{s} = \frac{\|\hat{\boldsymbol{\beta}}(s)\|_1}{\|\hat{\boldsymbol{\beta}}^0\|_1}$. O parâmetro $\tilde{s}$ tem a mesma vantagem que s', variar entre 0 e 1 e padronizar a penalização. Perceba que, em ambos os casos, depois de escolhido s' ou $\tilde{s}$, automaticamente também teremos s, uma vez que a relação entre eles é biunívoca. Neste trabalho, arbitrariamente, trabalharemos com a notação $\tilde{s}$.

Retornando à ideia da validação cruzada, o processo de estimação se repete para todos os grupos e, ao final, teremos um conjunto de erros de predição (EP) e erros quadráticos médios (EQM) para cada valor de $\tilde{s}$. A medida utilizada para determinar o parâmetro de ajuste ótimo é costuma ser o EQM, uma vez que seu cálculo é computacionalmente mais simples do que o do EP (CASAGRANDE, 2016). Assim, calcula-se a média dos valores de EQM para cada $\tilde{s}$ a fim de escolher o que apresente a menor média dos erros quadráticos médios.

Definido o valor mais conveniente para $\tilde{s}$, e consequentemente para s, estima-se novamente os coeficientes do LASSO com o banco de dados completo, a fim de ajustar o modelo final.

5.2.2 Graus de liberdade: uma outra maneira de estimar s

Para modelos aditivos, ou seja, do tipo $y_i = f(x_i) + \mathbf{e}_i$, com $i = 1, 2, ..., n$, $f(x_i)$ desconhecido e $\mathbf{e}_i \overset{iid}{\sim} (0,\ \sigma^2)$, os graus de liberdade (representados por df($\hat{y}$), do inglês *degrees of freedom*) são dados por:

$$df(\hat{y}) := \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{Cov}(\hat{\mathbf{y}}_\mathbf{i}, \mathbf{y}_\mathbf{i}). \tag{25}$$

Nesta expressão, a covariância tenta medir o grau de aleatoriedade que há

entre a variável resposta e o valor predito. Perceba que, quanto mais o modelo se adapta aos dados, maior será esta covariância, e consequentemente, maiores serão seus graus de liberdade. Se selecionarmos k variáveis para compor um modelo linear fixo, $\mathrm{df}(\hat{y}) = k$, já para modelos adaptativos, geralmente os graus de liberdade são maiores do que k. (HASTIE et al., 2015)

Porém, contrariando as expectativas sobre os graus de liberdade de modelos adaptativos, Zou et al. (2007) trazem um importante resultado quanto a estimação dos graus de liberdade do LASSO, "o número de parâmetros não-nulos é um estimador não viciado dos graus de liberdade do LASSO, e este resultado pode ser usado para construir um critério de seleção de modelos adaptativos que selecione eficientemente o ajuste ótimo do LASSO" [12](tradução nossa). Vale ressaltar que a demonstração deste resultado é válida para amostras finitas e matriz de delineamento de posto completo. Os autores também mostraram que este estimador é consistente.

Segundo Hastie et al. (2015) este comportamento do LASSO de selecionar variáveis reduzindo seus coeficientes a zero, também reflete na diminuição dos graus de liberdade deste modelo quando comparado às demais técnicas de seleção de modelos.

Os estudos de Zou et al. (2007) sobre os graus de liberdade do LASSO são importantes, pois podem representar mais uma maneira de estimar λ, porém com um esforço computacional menor. Sob esta ótica, seria suficiente calcular os estimadores de MQ e a partir deles e dos graus de liberdade, também seria possível encontrar o estimador LASSO ótimo.

[12] "the number of nonzero components of $\hat{\beta}$ is an exact unbiased estimate of the degrees of freedom of the LASSO, and this result can be used to construct adaptive model selection criteria for efficiently selecting the optimal LASSO fit"

6 GENERALIZAÇÕES E APLICAÇÕES DO LASSO

Neste trabalho direcionamos as atenções para a aplicação do LASSO em modelos de regressão lineares com variáveis contínuas, mas o alcance das aplicações da técnica não se limita apenas a essa classe de modelos.

O LASSO pode ser aplicado em regressões logísticas, e de maneira mais abrangente, em modelos lineares generalizados, além de modelos de Cox e estudos de análise de sobrevivência. Nas estimações, abordamos a estatística frequentista e demos uma abordagem padrão, que considera a distribuição dos erros normais e identicamente distribuídos, mas também é possível trabalhar com estimadores Bayesianos e técnicas de estatística não paramétrica. (HASTIE et al. 2015)

Além disso é importante destacar que várias técnicas estão sendo desenvolvidas, aproveitando parte das ideias aplicadas no LASSO, destacamos especialmente o *elastic net*. Desenvolvida por Zou e Hastie (2005, *apud* HASTIE et al., 2015), a ideia principal é combinar as penalizações L_1 e L_2, aproveitando boas características tanto do LASSO como da regressão em crista.

Outra proposta para trabalhar com variáveis correlacionadas é o *Group-lasso*, proposto por Yuan e Lin (2006, *apud* HASTIE et al., 2015). Assim como o *elastic net*, o objetivo também é trabalhar com grupos de variáveis, mas com o diferencial de poder incluir ou excluir todos de uma vez, ou determinar um coeficiente de regressão para o grupo inteiro. Sobre essa característica, Hastie et al. (2015) afirmam que "quando selecionamos variáveis para um modelo desse tipo, nós gostaríamos de incluir ou excluir grupos de uma vez, em vez de coeficientes individuais, e o *Group-lasso* é projetado para impor tal comportamento". [13](tradução nossa)

Apesar de desenvolvida em 1996, o LASSO é uma técnica relativamente nova se comparada com outras técnicas estatísticas que utilizamos. Esse caráter inovador é perceptível nas literaturas e publicações sobre o assunto, todas recentes e a maioria delas com discussões pouco exploradas. Muitas outras variações do LASSO foram propostas desde sua elaboração, e foge do escopo deste trabalho discuti-las individualmente. Mas para o leitor interessado sugerimos buscar maiores informações em Friedman et al. (2009), por exemplo. Os autores fizeram uma boa síntese das pesquisas e propostas mais recentes no âmbito discutido.

Na próxima seção discutiremos um pouco mais detalhadamente a ideia que embasa o *elastic net*, pois o pacote do *software* `R` que é utilizado nas estimações LASSO utiliza essa expressão.

[13] "when selecting variable for a such model we would typically want to include or exclude groups at a time, rather than individual coefficients , and the *Group-lasso* is designed to enforce such behavior."

6.1 *Elastic net*

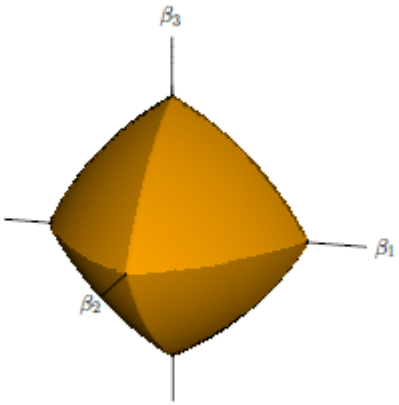
A técnica chamada *elastic net* também se baseia na ideia de estimação com penalização a fim de melhorar a acurácia dos estimadores MQ em situações adversas. A ideia é associar boas características do LASSO e da regressão em crista, estabelecendo a penalização a partir da equação (26):

$$\lambda\left[\frac{1}{2}(1-\alpha)\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1\right]. \tag{26}$$

Hastie et al. (2015) colocam que a principal motivação para esse estudo é o fato de o LASSO não ter a performance esperada quando as variáveis são muito correlacionadas. Adicionar a componente $\frac{1}{2}(1-\alpha)\|\boldsymbol{\beta}\|_2^2$ na penalização ajuda a controlar fortes correlações entre grupos de variáveis, mantendo a característica de tornar modelos esparsos que o LASSO possui. Em contrapartida, temos mais um parâmetro a estimar, α, que, grosseiramente, determina a proporção de cada penalização, em crista e LASSO, que será aplicada.

Geometricamente, também é possível perceber essas vantajosas características do *elastic net*, como mostra a Figura 15. "O gráfico do *elastic net* compartilha atributos do gráfico de L_1 e do gráfico de L_2: os cantos e bordas afiados encorajam a seleção, e o contorno curvo encoraja o compartilhamento de coeficientes"[14] (HASTIE et al., 2015, tradução nossa).

Figura 15 – *Elastic net* no $\mathbb{R}^3$



Fonte: Hastie et al. (2015).

Outra vantagem do *elastic net* é o fato de já estar implementado no `R`. Com o pacote `glmnet` é possível ajustar modelos usando o LASSO, a regressão em crista ou o *elastic net*. Discutiremos mais sobre esse pacote nas próximas seções, apresentando

[14] "the elastic net ball shares attributes of the l_2 ball and the l_1 ball: the sharp corners and edges encourage selection, and the curved contours encourage sharing of coeficients"

suas principais funções, argumentos e saídas, exemplificados a partir de dois conjuntos de dados.

6.2 Aplicação computacional do LASSO

A fim de equilibrar um pouco nossa discussão entre teoria e prática, uma vez que entendemos a importância de ambas na formação de um bom profissional, faremos uma discussão sobre as possibilidades de aplicação da técnica LASSO usando o `R`, que foi escolhido por se tratar de uma plataforma colaborativa e acessível, pois é gratuita e fácil de baixar, além de ser amplamente utilizada no meio acadêmico.

O primeiro pacote implementado no `R` que possibilitou um ajuste direto via LASSO foi o `lars`, acrônico de *Least Angle Regression*. Criado por Efron et al. (2003), foi definido pelos autores como um aperfeiçoamento da conhecida técnica de seleção de variáveis *Forward Selection*, fornecendo "dois algoritmos recentes promissores de construção de modelos de regressão lineares, LASSO e *Forward Stagewise*, motivados, em termos, por um método computacionalmente mais simples chamado Regressão de Ângulo Mínimo"[15] (EFRON et al.,2003, tradução nossa).

O `lars` possibilita a estimação dos coeficientes de diversos modelos, tantos quantos forem os valores atribuídos a λ, além da construção de gráficos que possibilitam a visualização dos caminhos percorridos por cada variável ao longo do processo de otimização, garantindo a principal característica do LASSO que é o potencial de zerar covariáveis e tornar o modelo esparso. Nesse pacote também foi implementado o método da validação cruzada para que o analista seja capaz de identificar o melhor valor de λ, e a partir dele escolher o modelo mais adequado para seu banco de dados.

Embora o pacote tenha suas indiscutíveis vantagens computacionais, Casagrande (2016) em seu estudo comparativo entre as técnicas de seleção e de encolhimento de modelos para situações com alta dimensionalidade e multicolinearidade, identificou que para dados simulados o LASSO apresentou um desempenho computacional razoável em termos de tempo, já para dados não simulados, com $n > p$ e multicolinearidade, o LASSO se apresentou pior. O pesquisador utilizou o pacote `lars` para o ajuste do lasso e analisou quatro técnicas: componentes principais, mínimos quadrados parciais, regressão em cristas e LASSO.

Em 2010, uma nova proposta de pacote foi desenvolvida e implementada no `R`. Criado por Friedman et al., o pacote `glmnet` utiliza um método de otimização chamado de *cyclical coordinate descent*, capaz de realizar as estimações ainda mais rápido. Outra vantagem do `glmnet` é que ele possibilita estimações via LASSO, regressão em crista e *elastic net*, essa última, conforme já discutido, é uma técnica que pretende aproveitar as

[15] "two promising recent model-building algorithms, the Lasso and Forward Stagewise linear regression, motivated in terms of a computationally simpler method called Least Angle Regression".

boas características da regressão em crista e do LASSO a fim de ajustar modelos mais parcimoniosos.

Por ser o pacote mais recente referente ao LASSO e por seu caráter mais abrangente, a partir daqui escolhemos trabalhar apenas com o pacote `glmnet`. Nos próximos tópicos aprofundaremos as discussões sobre as principais funções e comandos desse pacote, a medida que o relacionamos a um banco de dados como exemplo.

6.3 Ajustando o modelo com o pacote *glmnet*

O pacote `glmnet` foi desenvolvido para ajustar modelos, via máxima verossimilhança penalizada, usando a técnica *elastic net*, que, conforme já discutido, aplica a penalização de acordo com a expressão:

$$\lambda \left[\frac{(1-\alpha)}{2} \|\boldsymbol{\beta}\|_2^2 + \alpha \|\boldsymbol{\beta}\|_1 \right]. \tag{27}$$

Os parâmetros λ e α são responsáveis por controlar a força da penalidade e construir uma ponte entre a regressão em crista e o LASSO, respectivamente (FRIEDMAN et al., 2010). O primeiro é iterado quantas vezes se desejar, já o segundo é determinado pelo analista, variando no intervalo [0,1]. Se quiser ajustar via regressão em crista ou LASSO basta definir $\alpha = 0$ ou $\alpha = 1$, respectivamente. Para qualquer outro valor de α será ajustado um modelo combinando as duas técnicas

O pacote é capaz de trabalhar com modelos lineares, logísticos, multinomiais, Poisson e Cox, e seus coeficientes são estimados uma quantidade de vezes igual a λ. O padrão é que o `R` realize 100 iterações, mas este valor pode ser definido pelo pesquisador.

Ainda sobre o parâmetro λ, no próprio pacote já está implementado o comando para sua seleção por meio da validação cruzada, cuja quantidade de grupos utilizados também pode ser fixada. Ainda é possível determinar o melhor ou o pior valor de λ, utilizando o critério do menor erro quadrático médio, e fazer predições ou construir gráficos, por exemplo.

Por todas as discussões realizadas nos tópicos anteriores, é aconselhável trabalhar com as variáveis preditoras padronizadas e com variância 1, bem como é interessante que a variável resposta também esteja padronizada. No `R`, é possível realizar esta escolha para ambas as variáveis.

Podemos destacar algumas funções como principais para o ajuste do modelo desejado, são elas: `glmnet`,`print`, `plot`, `cv.glmnet`, `coef` e `predict`. A seguir daremos um breve resumo sobre alguns argumentos e respostas fundamentais dessas funções para uma análise inicial, por meio de uma aplicação. O exemplo escolhido foi retirado de Thomas[16] (1990 *apud* HASTIE et al. 2015), cujo banco de dados está disponível em `http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/`

[16]THOMAS,G.S. **The Rating Guide to Life in America's Small Cities**, Prometheus book.

`datasets/mlr/frames/frame.html`.

O exemplo discorre sobre as taxas de criminalidade em 50 cidades dos Estados Unidos e tem como variável resposta a taxa total global de crimes relatados por 1 milhão de habitantes. Foram consideradas 5 variáveis explicativas, cujo detalhamento está apresentado na Tabela 1

Tabela 1 – Variáveis analisadas no estudo sobre criminalidade nas cidades americanas.

Variável	Notação	Descrição
financiamento	x_1	Financiamento anual da polícia por residente (em dólares).
eb4	x_2	Percentual de pessoas com 25 anos ou mais que tenham cursado 4 anos de ensino médio.
noeb	x_3	Percentual de pessoas entre 16 e 19 anos que não estão e nem terminaram o ensino médio.
facul	x_4	Percentual de pessoas entre 18 e 24 anos que estão na faculdade.
x25facul	x_5	Percentual de pessoas com 25 anos ou mais com, no mínimo, 4 anos de faculdade cursados.

Fonte: Hastie et al. (2015)

Vale ressaltar que não é comum usar o LASSO para ajustar modelos com poucas variáveis, como é o caso do exemplo escolhido, geralmente a opção pela técnica se dá para resolver problemas com altas dimensões ($n \gg p$), ainda sim escolhemos trabalhar com esta ilustração por fins didáticos.

Para trabalhar com o pacote `glmnet` usamos os comandos a seguir para instalação e utilização das funções: "`install.packages(glmnet)`" e "`require(glmnet)`". Nos tópicos a seguir discutiremos as funções citadas bem como alguns argumentos e saídas principais, aplicados no banco de dados adotado. Destacamos ainda que será apresentado um panorama geral sobre as funções, para maiores detalhes sobre sintaxe e potencialidades do pacote o leitor pode consultar Friedman et al. (2010).

6.3.1 glmnet().

A função `glmnet` basicamente ajusta o modelo a partir de três argumentos principais: $\mathbf{X}$, y e α. O objeto $\mathbf{X}$ deve estar no formato de matriz e representa a matriz de especificação da análise, $\mathbf{y}$ é o vetor de respostas e α determina a técnica que deseja-se aplicar, como nosso foco é o LASSO, devemos usar $\alpha = 1$. É importante ressaltar que a função determina que sejam ajustados 100 modelos, um para cada valor de λ, mas, tanto o analista pode determinar um valor diferente, através do parâmetro `nlambda=`, como o próprio `R` interrompe as iterações para valores de λ menores que 100, se perceber que continuar o processo não está mais trazendo diferenças significativas no percentual dos

desvios explicados.

Outro argumento importante é o `family=`, serve para determinar o tipo de modelo que se deseja ajustar, apresentando as seguintes possibilidades: `gaussian`, `binomial`, `poisson`, `multinomial`, `cox` e `mgaussian`.

Através dos argumentos do `glmnet`, o analista também pode definir se padroniza ou não as variáveis preditoras e o vetor de respostas, se atribui pesos para as observações, se o intercepto deve ser ajustado, determinar o número máximo de variáveis que serão anuladas ou o número máximo de variáveis que deseja utilizar no modelo, dentre outras possibilidades.

O modelo ajustado em nosso exemplo foi chamado de `fit`, denominamos a matriz de especificação de **x** e o vetor de respostas de `taxacrime`. Uma vez ajustado o modelo, é possível calcular o intercepto (`fit\$$a_{0}$$`), os coeficientes para cada λ `(fit\$beta)`, os graus de liberdade para cada valor de λ `(fit\$df)`, a sequência de λ's utilizada `(fit\$lambda)`, dentre outros.

A seguir apresentamos os comandos discutidos aplicados no banco de dados que estamos utilizando como exemplo.

```
require(glmnet)
dados <- read.csv("data.csv",sep=";",dec=",", header=T, as.is=T)
attach(dados)
x=model.matrix(taxacrime~.,dados)[,-c(1)]
fit=glmnet(x,taxacrime,alpha=1,standardize = T,standardize.response = T)
fit$a0
fit$beta
fit$df
fit$lambda
```

6.3.2 print().

A função `print` responde três importantes informações a respeito dos modelos via `glmnet`, a quantidade de graus de liberdade, que, conforme discutido anteriormente é equivalente a quantidade de coeficientes não nulos, a porcentagem de explicação dos desvios e o valor de lambda para cada passo. Especificamente em nosso exemplo, uma parte da saída para o comando `print(fit)` é:

```
      Df    %Dev   Lambda
[1,]  0 0.00000 155.2000
[2,]  1 0.04827 141.4000
[3,]  1 0.08834 128.8000
[4,]  1 0.12160 117.4000
```

```
[5,]  1 0.14920 106.9000
[6,]  1 0.17220  97.4400
[7,]  1 0.19120  88.7800
[8,]  1 0.20700  80.9000
[9,]  1 0.22010  73.7100
[10,]  1 0.23100  67.1600
```

6.3.3 plot().

A função `plot`, no geral, produz gráficos. No contexto de nosso estudo, aplicado a função `glmnet`, os gráficos elaborados relacionam cada variável com o conjunto de valores de λ, o número de variáveis selecionadas, e o valor que o coeficiente relativo a cada variável assume. A Figura 16, obtida a partir do comando `plot(fit)`, mostra a aplicação desta função no modelo estimado no tópico anterior para o exemplo que estamos estudando.

No eixo horizontal temos o valores que λ assumiu ao longo dos passos. Cada curva do gráfico representa uma variável preditora, perceba que todas elas iniciam com coeficiente igual a zero, e a medida que o valor de λ aumenta seus coeficientes também aumentam. Neste gráfico podemos perceber claramente o poder de "encolhimento"que o LASSO possui, por exemplo, se fosse escolhido $\lambda = 10$, apenas duas variáveis teriam coeficiente diferente de zero, o que é equivalente a dizer que apenas estas foram selecionadas para o modelo. A quantidade de variáveis selecionadas para cada λ está representada acima do gráfico.

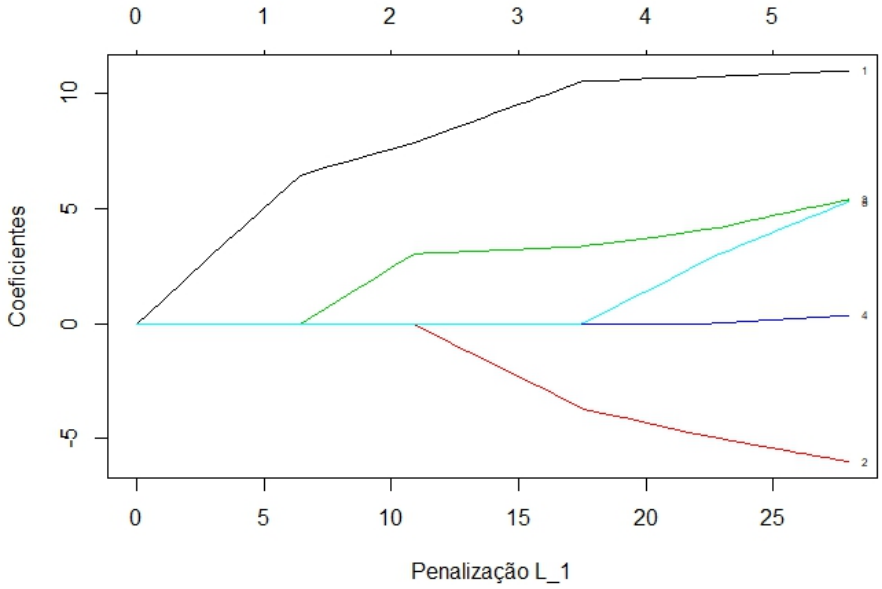
6.3.4 cv.glmnet().

Para que a técnica LASSO possa auxiliar na seleção de variáveis, tornando possível a construção de um bom modelo, ressaltamos mais uma vez a importância do parâmetro λ, pois é a partir do valor determinado para ele que os demais coeficientes do modelo serão estimados.

A função que será utilizada neste processo é `cv.glmnet()`, criada para aplicar a técnica da validação cruzada nas funções ajustadas via `glmnet`, fornecer o melhor λ e produzir gráficos. Os argumentos principais são a matriz de especificação, `x`, o vetor de respostas e o α, que continuaremos adotando como 1. Também é possível determinar o número de grupos em que a validação cruzada vai dividir o banco de dados, através do argumento `nfolds=`, o padrão do `R` é 10, e o valor mínimo que o analista pode escolher é 3, podendo chegar a atingir um valor igual ao número de observações, nesse último caso temos uma técnica conhecida como validação cruzada *leave-one-out*. Os demais argumentos são os mesmos do `glmnet`.

Dos objetos que a função pode responder, destacamos: a sequência de lamb-

Figura 16 – Representação gráfica da relação entre λ e a quantidade de coeficientes no exemplo sobre a criminalidade nos EUA



Fonte: Autora

das utilizada (`cv.lasso\$lambda`), o erro médio de validação cruzada para cada um deles(`cv.lasso\$cvm`), o desvio-padrão de cada λ (`cv.lasso\$cvsd`), o λ que apresenta o menor erro (`cv.lasso\$lambda.min`) e o maior valor de λ cujo o erro está a um erro padrão do mínimo(`cv.lasso\$lambda.1se`). Aplicamos a função `cv.glmnet()` em nosso exemplo com o nome de `cv.lasso`. Alguns dos comandos mais importantes descritos aqui e suas saídas serão apresentados a seguir.

```
#Comando aplicado ao exemplo da taxa de criminalidade nas cidades americanas
cv.lasso=cv.glmnet(x,taxacrime, alpha=1)
#Sequência de lambdas
> cv.lasso$lambda
[1] 155.1523086 141.3690051 128.8101723 117.3670316 106.9404680  97.4401716
[7]  88.7838553  80.8965423  73.7099164  67.1617306  61.1952676  55.7588487
[13]  50.8053862  46.2919757  42.1795242  38.4324115  35.0181820  31.9072632
[19]  29.0727099  26.4899705  24.1366745  21.9924389  20.0386912  18.2585091
[25]  16.6364735  15.1585350  13.8118926  12.5848821  11.4668759  10.4481902
[31]   9.5200018   8.6742711   7.9036727   7.2015322   6.5617680   5.9788386
[37]   5.4476950   4.9637368   4.5227721   4.1209815   3.7548849   3.4213112
[43]   3.1173714   2.8404327   2.5880965   2.3581771   2.1486832   1.9578001
[49]   1.7838745   1.6254000   1.4810040   1.3494357   1.2295555   1.1203252
[55]   1.0207986   0.9301137   0.8474849   0.7721967   0.7035969
```

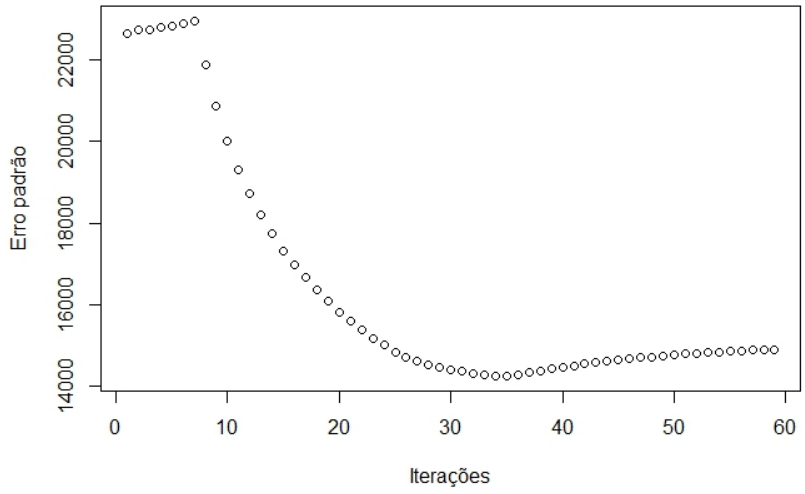
```
# Sequência de erros
> cv.lasso$cvm
[1] 88298.56 87540.49 87005.87 85491.68 84166.97 82393.09 80218.77 78455.48
[9] 77008.34 75840.88 74821.96 73956.91 73169.99 72450.09 71818.90 71267.24
[17] 70733.81 70258.33 69841.22 69566.60 69539.71 69527.23 69544.43 69585.54
[25] 69644.43 69727.06 69922.43 70198.82 70479.59 70768.52 71051.26 71319.34
[33] 71571.86 71810.05 72021.83 72196.12 72361.27 72545.47 72742.49 72925.09
[41] 73094.46 73251.36 73396.20 73529.84 73653.24 73764.98 73863.74 73959.04
[49] 74066.42 74170.89 74267.58 74358.87 74443.98 74522.59 74594.22 74658.23
[57] 74714.79 74766.91 74815.15

# Sequência de erros padrão
> cv.lasso$cvsd
[1] 26808.66 26959.15 27098.07 26689.94 26333.37 25550.16 24325.47 23285.00
[9] 22415.61 21698.73 21106.45 20641.70 20266.63 19965.36 19728.06 19533.67
[17] 19385.87 19276.09 19195.76 19162.01 19210.17 19278.91 19353.30 19430.92
[25] 19509.92 19594.58 19706.86 19851.40 19986.50 20111.02 20228.43 20339.55
[33] 20444.16 20542.57 20636.08 20727.56 20813.27 20900.34 20988.17 21068.46
[41] 21142.28 21210.06 21271.67 21328.16 21380.12 21428.11 21472.89 21518.02
[49] 21573.26 21627.41 21677.17 21722.02 21761.73 21798.53 21832.34 21863.63
[57] 21891.76 21918.25 21942.67

#Lambdas especiais
> cv.lasso$lambda.min
[1] 21.99244
> cv.lasso$lambda.1se
[1] 155.1523

# Quantidade de variáveis selecionadas para cada iteração
> cv.lasso$nzero
s0  s1  s2  s3  s4  s5  s6  s7  s8  s9 s10 s11 s12 s13 s14 s15 s16 s17 s18
0   1   1   1   1   1   1   1   1   1   2   2   2   2   3   3   3   3   3
s19 s20 s21 s22 s23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35 s36 s37
3   3   3   3   3   3   3   3   3   3   3   4   4   4   4   4   4   4   5
s38 s39 s40 s41 s42 s43 s44 s45 s46 s47 s48 s49 s50 s51 s52 s53 s54 s55 s56
5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
s57 s58
5   5
```

Perceba que, embora o padrão seja $\lambda = 100$, em nosso ajuste só foram realizadas 64 iterações, pois o `R` interrompe o processo quando a diferença entre o percentual explicado dos desvios de um passo para o seguinte for menor do que um valor de referência que indica convergência. Especificamente em nosso exemplo, o `R` ajustou 64 modelos.

Dois gráficos que podem ser úteis na análise das iterações de λ bem como em sua escolha estão exemplificados nas Figuras 17 e 18. O primeiro auxilia na observação do comportamento dos erros padrão ao longo das iterações, em nosso exemplo foi obtido a partir do comando `plot(cv.lasso\$cvsd)`.

Já o segundo apresenta a curva da validação cruzada (*cross validation curve*) para o nosso exemplo. Este gráfico é muito elucidativo, pois nos mostra o comportamento do erro quadrático médio ao longo das iterações, e esta medida é a que usamos para escolher o λ ótimo que será usado no ajuste do modelo final. Para determiná-lo usamos o comando `cv.lasso\$lambda.min`, que responderá o valor desejado. Em nosso exemplo, esse valor é 24.13667, está representado no gráfico por uma linha pontilhada vertical e a partir de agora o denotaremos por `bestlam`.

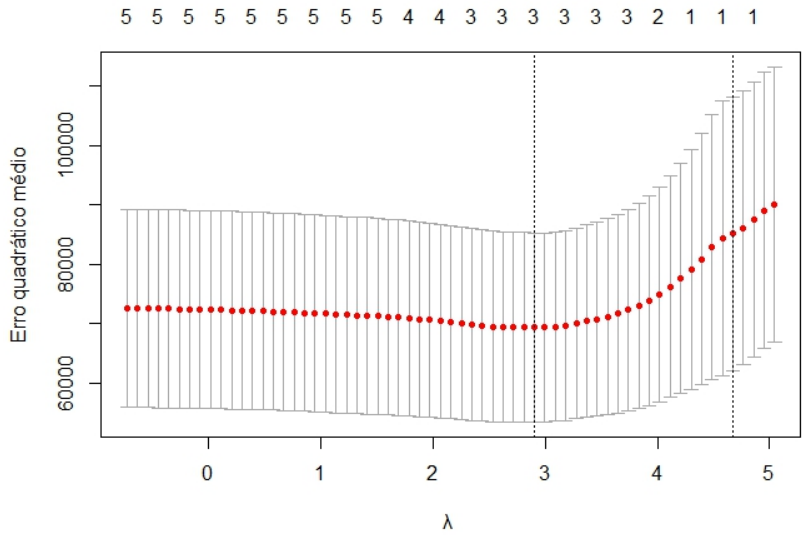
Figura 17 – Comportamento dos desvios-padrões ao longo das iterações no exemplo sobre criminalidade nas cidades americanas.



Fonte: Autora

Além disso, com a análise da Figura 18, também é possível saber quantas variáveis não nulas existem no ajuste para cada λ, essa informação é interessante para nós uma vez, para o λ escolhido, essas são as variáveis que efetivamente serão utilizadas no modelo final. Para nosso exemplo, 3 variáveis foram consideradas significativas, no próximo tópico vamos identificá-las e ajustar o modelo final.

Figura 18 – Curva de validação cruzada no exemplo sobre criminalidade nas cidades americanas.



Fonte: Autora

6.3.5 coef().

A saída do comando `coef` são os coeficientes do modelo para cada λ. Raramente essa informação é relevante para o analista, afinal seu interesse está em um modelo específico. Uma vez que nosso parâmetro de escolha é o melhor valor de λ, e este já foi determinado, vamos buscar quais as variáveis foram consideradas significativas e seus respectivos coeficientes. Para o exemplo que estamos estudando, obtivemos o seguinte resultado:

```
> bestlam=cv.lasso$lambda.min
> coef(cv.lasso, s=bestlam)

(Intercept)   445.373756
financiamento   9.495263
eb4            -2.304032
noeb            3.215727
facul           .
X25facul    .
```

Apenas alguns coeficientes do modelo e o intercepto foram determinados, então podemos concluir que as variáveis selecionadas pelo LASSO são apenas as que apresentam coeficiente. Em nosso exemplo as escolhidas serão: **financiamento anual da polícia por residente(em dólares)**, **percentual de pessoas com 25 anos ou mais que tenham cursado 4 anos de ensino médio** e o **percentual de pessoas entre 16 e**

19 anos que não estão e nem terminaram o ensino médio. Assim, entendemos que essas são as características que mais influenciam na As demais variáveis foram "zeradas", portanto não aparecerão em nosso modelo.

Assim, temos que o modelo final, estimado via LASSO, foi:

$$\hat{y}_i = 445,37 + 9,49 \cdot x_{1i} + 3,21 \cdot x_{3i} - 2,3 \cdot x_{5i} \tag{28}$$

Quanto ao erro padrão, quando ajusta-se modelos tradicionais, através da função `lm()`, o `R` apresenta esses valores para cada coeficiente, e essas medidas são muito importantes no momento de fazer o diagnóstico do modelo. Tibshirani (1996) ressalta a dificuldade de realizar o cálculo desta medida para os estimadores LASSO, "como o estimador LASSO é uma função não linear e não diferenciável em seus valores de resposta, mesmo para um valor fixo de *lambda*, é difícil obter uma estimativa precisa de cada erro padrão, uma abordagem possível é via *Bootstrap*"[17] (tradução nossa). O pacote `glmnet` não possui comando específico para determinar o erro padrão dos estimadores, e neste trabalho, não entraremos nas discussões da aplicação do *Bootstrap* no LASSO.

6.3.6 predict().

O comando `predict()` é utilizado para fazer predições tanto em modelos ajustados via `glmnet` como usando o `cv.glmnet`. Os argumentos básicos que devem ser determinados são: o modelo que se deseja utilizar, uma nova matriz de predições, cujo argumento é `newx=` e um valor fixo para o λ, através do argumento `s=`. Vale ressaltar que o analista pode determinar mais de um valor para `s`, neste caso a saída será uma matriz, na qual cada elemento a_{ij} será o valor predito do indivíduo i para $s = j$.

Um outro argumento que pode ser útil é o `type=`, esse pode ser igualado a "`response`", "`coefficients`"ou "`nonzero`", que determinam respectivamente, o $\hat{y}$ para cada indivíduo, as variáveis selecionadas e a quantidade de variáveis selecionadas. Se o analista não especificar este argumento, o padrão do `R` é determinar os valores preditos da variável resposta.

Em nosso exemplo, ajustamos uma nova matriz de especificação apenas com as variáveis selecionadas, um novo modelo com essas mesmas variáveis, fixando o valor de λ igual ao `bestlam` e usamos a função `predict()` para determinar os valores preditos das dez primeiros indivíduos. Os comandos utilizados e a saída do `R` estão exemplificados a seguir:

```
> x2=model.matrix(taxacrime~.,dados)[,-c(1,5,6)]
> model=glmnet(x2,taxacrime, alpha=1, standardize = T, standardize.response = T)
```

[17] "since the LASSO estimate is a non linear and non differentiable function of the response values even for a fixed value of *lambda*, it is difficult to obtain an accurate estimate of its standard errors. One approach is via *Bootstrap*."

```
> predict(model,newx = x2[1:10,],s=bestlam)
1  690.0589
2  618.7049
3  883.2046
4  611.5136
5  944.6076
6  551.8070
7  650.1273
8  657.6719
9  650.7351
10 616.6023
```

Nos tópicos anteriores, apresentamos as principais funções para estimação de modelos de regressão via LASSO. O pacote `glmnet` possui muitas outras possibilidades que podem ser encontradas em Friedman et al. (2010).

A seguir apresentaremos outro exemplo e a partir deste faremos uma comparação entre o LASSO e algumas técnicas de seleção de variáveis.

6.4 Comparação entre técnicas de seleção de variáveis.

Neste tópico serão feitas comparações entre algumas das técnicas de seleção de variáveis mais conhecidas, a partir de uma aplicação. Desde já é importante ressaltar que modelos são aproximações, logo os resultados que encontraremos sempre serão estimativas, jamais "verdades absolutas". Assim, as técnicas podem apresentar resultados um pouco diferentes, apontando quantidades ou selecionando variáveis distintas, cabe ao pesquisador buscar mecanismos que auxiliem na escolha do modelo final.

Nosso estudo será baseado no banco de dados denominado `Hitters`, disponível em James et al (2013) e disponibilizado pelos autores no `R`, sendo necessário apenas baixar o pacote `ISLR`, no qual os dados já estão implementados.

O interesse desse estudo é construir um modelo cuja variável resposta (`Salary`) é o salário de rebatedores que jogam na *Major League Baseball* (MLB), a maior organização de beisebol dos Estados Unidos da América. Para tanto são analisadas 19 variáveis e 322 observações, referentes a temporada de 1986, a fim de fazer inferências sobre os valores de seus salários para a temporada de 1987. Como existem dados omissos no banco de dados, esses foram retirados e trabalharemos apenas com as 263 que apresentam informações para todas a s variáveis . A Tabela 2 apresenta um breve resumo das variáveis em questão:

Tabela 2 – Variáveis analisadas no banco de dados Hitters.

Variável	Notação	Descrição
AtBat	x_1	Número de vezes com o bastão em 1986
Hits	x_2	Número de *hits* em 1986
HmRun	x_3	Número de *home runs* em 1986
Runs	x_4	Número de *runs* em 1986
RBI	x_5	Número de *runs batted* em 1986
Walks	x_6	Número de *walks* em 1986
Years	x_7	Número de anos na MBL
CatBat	x_8	Número de vezes com o bastão na carreira
CHits	x_9	Número de *hits* na carreira
CHmRun	x_{10}	Número de *home runs* na carreira
CRuns	x_{11}	Número de *runs* na carreira
CRBI	x_{12}	Número de *runs batted* na carreira
CWalks	x_{13}	Número de *walks* na carreira
League	x_{14}	Fator que indica a Liga do jogador no final de 1986
Division	x_{15}	Fator que indica a Divisão do jogador no final de 1986
PutOuts	x_{16}	Número de *put outs* em 1986
Assists	x_{17}	Número de *assists* em 1986
Errors	x_{18}	Número de erros em 1986
NewLeague	x_{19}	Fator que indica a Liga do jogador no início de 1986

Fonte: James *et al.* (2013)

6.4.1 Métodos *Subset selection*

Como já discutimos anteriormente, técnicas deste tipo selecionam um subconjunto de variáveis que julga mais influente sobre a variável resposta e ajusta o modelo a partir dessa escolha.

No caso do *best subset* são ajustados todos os modelos possíveis e, a partir de algum critério estipulado pelo pesquisador, seleciona-se as variáveis e estima-se os parâmetros. Para tanto, utilizaremos a função `regsubsets()`, presente na biblioteca `leaps`. Quando aplicamos a função `summary` ao modelo estimado via `regsubsets()`, a saída do R determina, através de um asterisco, quais variáveis devem figurar no modelo a depender da quantidade de variáveis escolhidas. Assim, por exemplo, se o pesquisador desejar estimar o modelo com três variáveis, as escolhidas seriam: `CRBI`, `Hits` e `PutOuts`.

```
bestsubsetfit=regsubsets(Salary~.,Hitters)
summary(bestsubsetfit)

Selection Algorithm: exhaustive
AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns
1 " "   " "  " "   " "  " " " " " "   " "   " "    " "   " "    " "
```

```
2 " "   "*"  " "   " "  " " " " "   " "   " "    " "   " "    " "
3 " "   "*"  " "   " "  " " " " "   " "   " "    " "   " "    " "
4 " "   "*"  " "   " "  " " " " "   " "   " "    " "   " "    " "
5 "*"   "*"  " "   " "  " " " " "   " "   " "    " "   " "    " "
6 "*"   "*"  " "   " "  " " "*"   " "   " "    " "   " "    " "
7 " "   "*"  " "   " "  " " "*"   " "   "*"    "*"   "*"    " "
8 "*"   "*"  " "   " "  " " "*"   " "   " "    " "   "*"    "*"
CRBI CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
1 "*"  " "    " "      " "       " "      " "     " "    " "
2 "*"  " "    " "      " "       " "      " "     " "    " "
3 "*"  " "    " "      " "       "*"      " "     " "    " "
4 "*"  " "    " "      "*"       "*"      " "     " "    " "
5 "*"  " "    " "      "*"       "*"      " "     " "    " "
6 "*"  " "    " "      "*"       "*"      " "     " "    " "
7 " "  " "    " "      "*"       "*"      " "     " "    " "
8 " "  "*"    " "      "*"       "*"      " "     " "    " "
```

O padrão do `R` é analisar apenas as 8 variáveis mais correlacionadas com a variável resposta, mas caso o analista tenha interesse, é possível analisar quantas variáveis seja desejado, através do parâmetro `nvmax=`.

É possível repetir o mesmo procedimento para as técnicas *Forward Stepwise* e *Backward Stepwise*, basta utilizar o parâmetro `method=` na função `regsubsets` e determinar qual método deseja, como mostra o exemplo a seguir.

```
fwdfit=regsubsets(Salary~.,data=Hitters, method = "forward")
summary(fwdfit)
```

```
    AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns
1    " "   " "  " "   " "  " " " " " "   " "   " "    " "   " "    " "
2    " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
3    " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
4    " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
5    "*"   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
6    "*"   "*"  " "   " "  " " " " "*"   " "   " "    " "   " "    " "
7    "*"   "*"  " "   " "  " " " " "*"   " "   " "    " "   " "    " "
8    "*"   "*"  " "   " "  " " " " "*"   " "   " "    " "   " "    "*"
    CRBI CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
1    "*"  " "    " "      " "       " "      " "     " "    " "
2    "*"  " "    " "      " "       " "      " "     " "    " "
3    "*"  " "    " "      " "       "*"      " "     " "    " "
```

```
4    "*"  " "    " "     "*"       "*"     " "     " "    " "
5    "*"  " "    " "     "*"       "*"     " "     " "    " "
6    "*"  " "    " "     "*"       "*"     " "     " "    " "
7    "*"  "*"    " "     "*"       "*"     " "     " "    " "
8    "*"  "*"    " "     "*"       "*"     " "     " "    " "
```

```
bckfit=regsubsets(Salary~.,data=Hitters, method ="backward")
summary (bckfit)
```

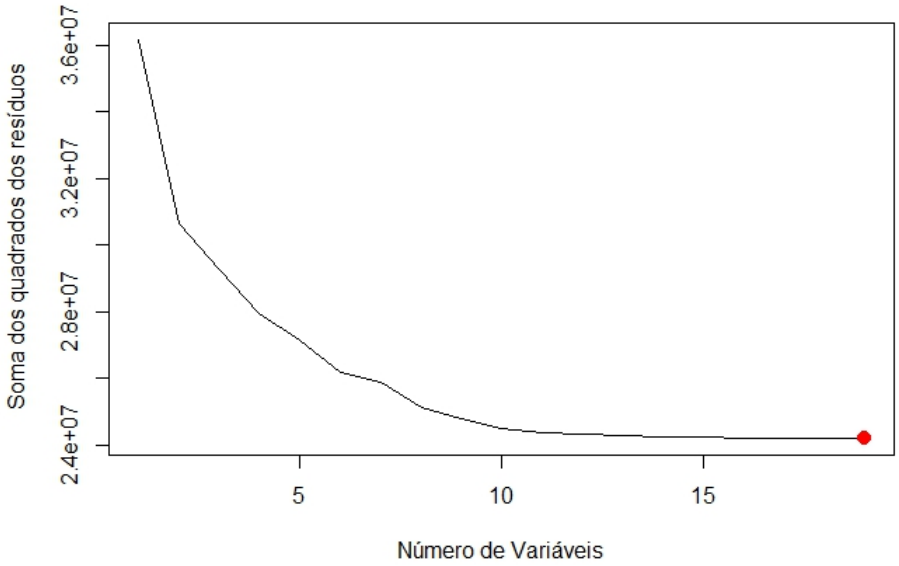
```
   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns
1   " "   " "  " "   " "  " " " " " "   " "    " "    " "   " "    "*"
2   " "   "*"  " "   " "  " " " " " "   " "    " "    " "   " "    "*"
3   " "   "*"  " "   " "  " " " " " "   " "    " "    " "   " "    "*"
4   "*"   "*"  " "   " "  " " " " " "   " "    " "    " "   " "    "*"
5   "*"   "*"  " "   " "  " " " " "*"   " "    " "    " "   " "    "*"
6   "*"   "*"  " "   " "  " " " " "*"   " "    " "    " "   " "    "*"
7   "*"   "*"  " "   " "  " " " " "*"   " "    " "    " "   " "    "*"
8   "*"   "*"  " "   " "  " " " " "*"   " "    " "    " "   " "    "*"
```

```
    CRBI CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
1   " "  " "    " "     " "       " "     " "     " "    " "
2   " "  " "    " "     " "       " "     " "     " "    " "
3   " "  " "    " "     " "       "*"     " "     " "    " "
4   " "  " "    " "     " "       "*"     " "     " "    " "
5   " "  " "    " "     " "       "*"     " "     " "    " "
6   " "  " "    " "     "*"       "*"     " "     " "    " "
7   " "  "*"    " "     "*"       "*"     " "     " "    " "
8   "*"  "*"    " "     "*"       "*"     " "     " "    " "
```

Podemos perceber que as técnicas não se comportam da mesma maneira, selecionando conjuntos distintos de variáveis, portanto é preciso estipular métodos e critérios que auxiliem na escolha de um bom modelo. Analisando 4 desses possíveis critérios: a soma dos quadrados dos resíduos, o coeficiente de determinação ajustado, a distância C_p e o BIC, as Figuras 19, 20, 21 e 22 mostram seus comportamentos a medida que a quantidade de variáveis aumenta, marcando com um ponto vermelho o valor ideal para esta quantidade em cada um dos critérios, considerando o método *best subset*. A quantidade de variáveis selecionadas para cada um dos critérios é 19, 11, 8 e 6, respectivamente. Neste caso, seria aconselhável escolher o critério BIC, uma vez que este aponta a menor quantidade de características a serem analisadas e quanto menor for este valor, mais fácil

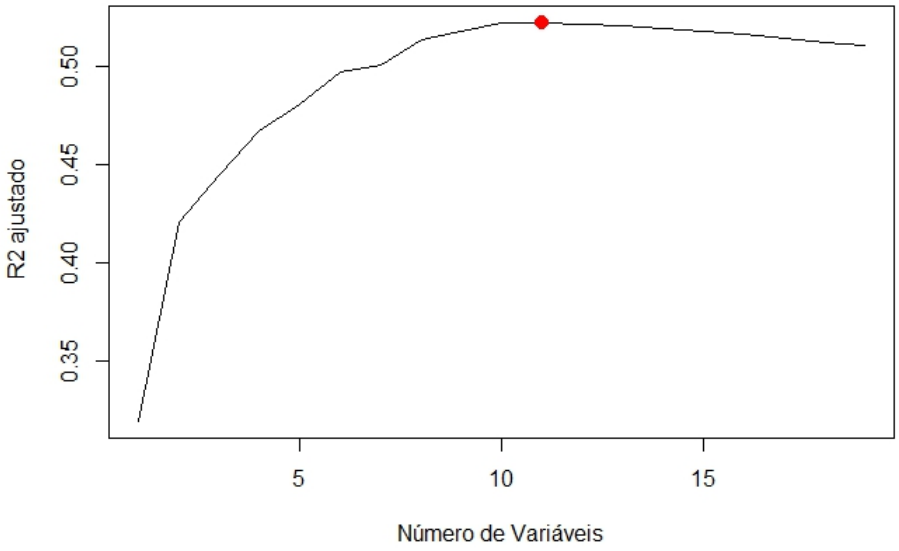
fica a interpretação.

Figura 19 – Número de variáveis *versus* comportamento da Soma dos Quadrados dos Resíduos no banco de dados Hitters.



Fonte: Autora

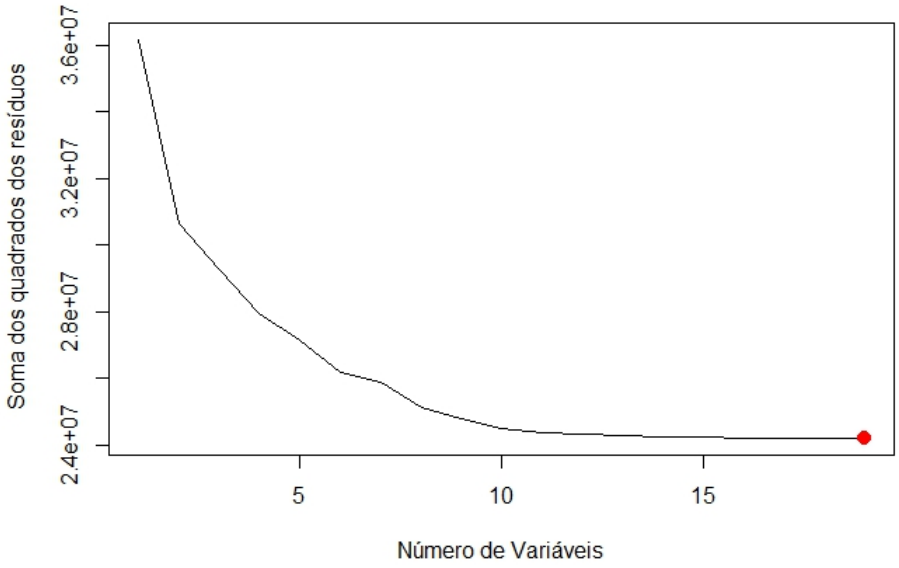
Figura 20 – Número de variáveis *versus* comportamento do Coeficiente de Determinação ajustado no banco de dados Hitters.



Fonte: Autora

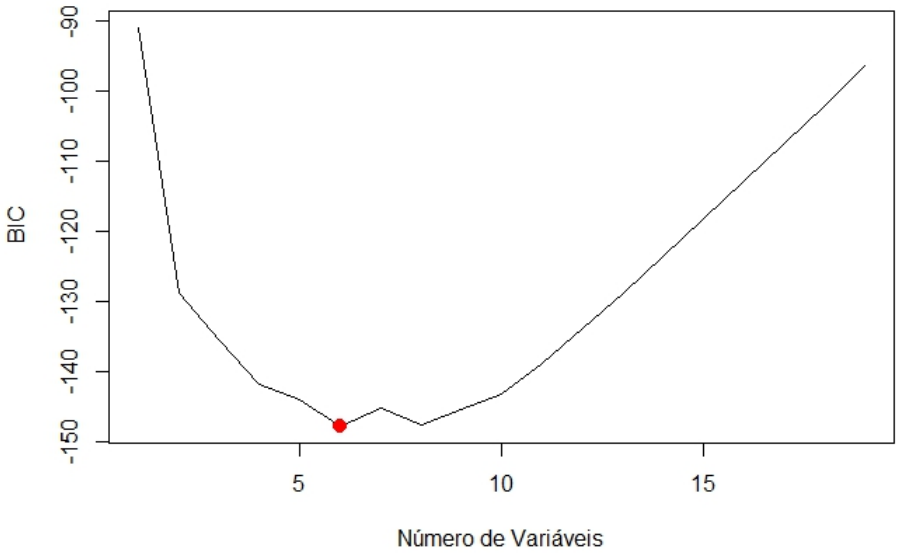
Usando a mesma ideia para os outros dois métodos, a quantidade de variáveis selecionadas por cada critério permanece a mesma, a única exceção é o critério BIC na

Figura 21 – Número de variáveis *versus* comportamento da estatística C_p no banco de dados Hitters.



Fonte: Autora

Figura 22 – Número de variáveis *versus* comportamento da estatística C_p no banco de dados Hitters.



Fonte: Autora

técnica *Backward Stepwise*, que ao invés de 6 variáveis, sugere o uso de 8. Ainda sim, este critério continua sendo o que aponta a menor quantidade de variáveis.

Para que a escolha possa ser mais consistente, baseando-se também em testes de erros, James et al. (2013) propõe mais duas possibilidades de técnicas para determinar

a quantidade de variáveis a serem escolhidas: o *Validation set approach* e o método da validação cruzada. De maneira geral, os dois se baseiam na divisão do banco de dados em grupos de teste e de treinamento, o que possibilita uma comparação entre resultados reais e estimados e pode dar ao pesquisador uma noção da amplitude dos erros que seu modelo está gerando. Para este exemplo, o autor constatou que as referidas técnicas sugerem o uso de 10 e 11 variáveis, respectivamente.

A Tabela 3 compila quais variáveis foram escolhidas para cada uma das técnicas aplicadas até então.

Tabela 3 – Variáveis selecionadas por cada uma das técnicas analisadas.

Técnica	**Número de variáveis**	**Variáveis**
Best subset (BIC)	6	`CRBI`, `Hits`, `PutOuts`, `DivisionW`, `AtBat` e `Walks`.
Forward Stepwise (BIC)	6	`CRBI`, `Hits`, `PutOuts`, `DivisionW`, `AtBat` e `Walks`.
Backward Stepwise(BIC)	8	`CRBI`, `Hits`, `PutOuts`, `AtBat`, `Walks`, `DivisionW`, `CWalks` e `CRBI`.
Validation set approach	10	`Hits`, `Walks`, `CRuns`, `CRBI`, `PutOuts`, `Assists`, `CatBat`, `CWalks`, `AtBat` e `DivisionW`.
Validação cruzada	11	`LeagueN`, `Hits`, `Walks`, `CRuns`, `CRBI`, `PutOuts`, `Assists`, `CatBat`, `CWalks`, `AtBat` e `DivisionW`.

Fonte: Autora

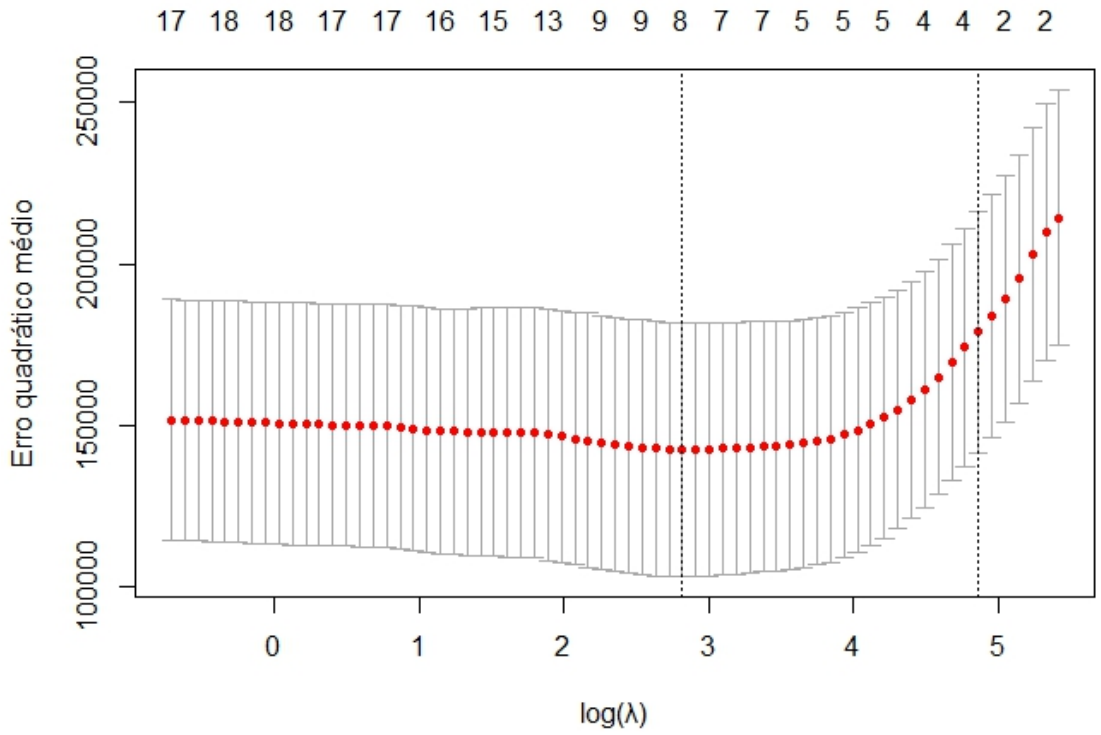
6.4.2 Métodos *Shrinkage*

Neste tópico buscaremos identificar como se comporta a regressão em crista e o LASSO aplicados no exemplo em discussão. Usando as funções `glmnet` e `cv.glmnet`, James et al. (2013) usa a regressão ridge para estimar os coeficientes de regressão do modelo, usando a validação cruzada para determinar o melhor parâmetro de penalização. Conforme já discutimos ao longo deste trabalho a regressão ridge não zera nenhum dos coeficientes, embora, aqueles relacionados à variáveis que não são tão significativas assumam valores muito próximos a zero. De qualquer maneira o modelo será ajustado considerando todas as variáveis, o que signifca que a regressão em crista não configura-se como uma técnica de seleção de variáveis.

Quanto ao LASSO, usando a função `cv.glmnet`, traçamos a curva de validação

cruzada, na Figura 23, a fim de identificar o comportamento de λ ao longo das iterações e buscar melhor valor para este parâmetro. A partir deste gráfico já podemos suspeitar que os graus de liberdade do modelo estão entre 7 e 8, já que pelo gráfico apresenta o menor erro quadrático, e consequentemente esta deve ser a quantidade de variáveis selecionada pela técnica.

Figura 23 – Curva de validação cruzada do banco de dados Hitters.



Fonte: Autora

Após definido o melhor valor para o parâmetro de penalização, $\lambda = 16,78$, estimamos novamente os coeficientes do regressão para esse λ específico, desta vez considerando o banco de dados completo. Através do comando `coef()` encontramos que a técnica LASSO selecionou 7 variáveis explicativas: `LeagueN`, `Walks`, `Hits`, `CRBI`, `PutsOuts`, `CRuns` e `DivisionW`, e seus respectivos coeficientes, conforme saída do `R` apresentada a seguir.

```
xlasso=model.matrix(Salary~.,Hitters)[,-c(1)]
ylasso=Hitters$Salary
lassofinal=glmnet(xlasso,ylasso, lambda = grid)
coef(lassofinal, s=bestlam2)

(Intercept)   18.5394844
AtBat          .
Hits           1.8735390
```

```
HmRun           .
Runs            .
RBI             .
Walks           2.2178444
Years           .
CAtBat          .
CHits           .
CHmRun          .
CRuns           0.2071252
CRBI            0.4130132
CWalks          .
LeagueN         3.2666677
DivisionW    -103.4845458
PutOuts         0.2204284
Assists         .
Errors          .
NewLeagueN      .
```

Então, o modelo ajustado via LASSO é dado por:

$$\hat{y}_i = 18,54 + 1,82 \cdot x_{2,i} + 2,22 \cdot x_{6,i} + 0,21 \cdot x_{11,i} + 0,41 \cdot x_{12,i} + 3,27 \cdot x_{14,i} - 103,49 \cdot x_{15,i} + 0,22 \cdot x_{16,i}$$

A técnica LASSO apresentou uma quantidade menor de variáveis selecionadas, se comparada as técnicas de seleção de subconjuntos, exceto quando considera-se a análise direta do BIC para as técnicas *Best subset* e *Forward Stepwise*. Ainda sim, podemos considerar o LASSO superior, pois essas técnicas não fazem nenhum tipo de análise de erro, diferentemente do LASSO que aplica a validação cruzada e é capaz de fazer a comparação, pelo menos entre os grupos de teste e de treinamento.

Finalizamos essa discussão ressaltando que outras técnicas de diagnóstico poderiam ser utilizadas para julgar a acurácia deste modelo, inclusive acerca dos cálculos de erros padrão e intervalos de confiança para os estimadores, no entanto, seriam necessárias maiores investigações e técnicas de reamostragem, como por exemplo, o *Bootstrap*, ficando esta discussão pendente para próximos trabalhos.

7 CONCLUSÕES E PESQUISAS FUTURAS

O volume de dados com os quais analistas e pesquisadores trabalham está a cada dia maior, ainda mais com os crescentes avanços da tecnologia disponível para essas pesquisas. Com isso áreas de estudo como *Machine Learning* e *Statistical Learning* ganham ainda mais destaque, e a técnica LASSO pode ser muito útil neste contexto, o que torna sempre bem-vindos estudos e discussões sobre a mesma

Em nosso trabalho buscamos apresentar a referida técnica para ser usada na estimação de modelos lineares, uma vez que o MMQ não se mostre adequado, e com isso apresentar as principais vantagens do LASSO: a melhora na acurácia das previsões e a possibilidade de facilitar a interpretação de modelos, em razão de sua habilidade em ajustar modelos esparsos, revelando outra importante faceta dessa técnica, comportar-se como uma possibilidade para problemas de seleção de variáveis, especialmente em bancos de dados com alta dimensionalidade.

Outra grande vantagem do LASSO é o fato de ser uma técnica que utiliza penalizações na estimação de seus parâmetros, a partir de regiões de restrição convexas, característica que facilita o processo de estimação.

Também discutimos os aspectos teóricos relacionados à estimação dos parâmetros do modelo via LASSO, tanto os coeficientes de regressão como um novo parâmetro de penalização que surge para tentar diminuir a variância de situações em que há multicolinearidade ou alta dimensionalidade. Discutimos também algumas técnicas mais recentes que despontam na literatura, com o objetivo de suprir algumas desvantagens que o LASSO apresenta.

Finalizamos as discussões abordando, através do *software* `R`, os pacotes `glmnet` e `cv.glmnet` e suas principais funções, a partir de dois exemplos. No segundo, podemos ainda comparar o LASSO com outras técnicas de seleção de variáveis já amplamente discutidas pela literatura.

Ao longo de todo trabalho, podemos perceber que as refêrencias e as pesquisas que envolvem o LASSO são bem atuais e estão em pleno desenvolvimento, logo este trabalho se faz relevante uma vez que se propõe a discutir, compilar e tornar um pouco mais elucidativas as questões referentes à técnica, especialmente porque os trabalhos em língua portuguesa são bem escassos.

Nossa discussão foi introdutória, evidenciando que ainda há muito o que desenvolver, tanto no campo teórico quanto prático, no que diz respeito ao LASSO. Aplicações da técnica em outros modelos, como os lineares generalizados, análises a partir da estatística bayesiana, análises de erro padrão e intervalos de confiança para estimadores e outras técnicas de diagnóstico, além de aplicações em bancos de dados maiores, podem ser temas de trabalhos e discussões futuras.

REFERÊNCIAS

BÜHLMANN, P.; VAN DE GEER, S. **Statistics for high dimensional data:** *methods, theory and applications.* Berlim: Springer, 2011.

CASAGRANDE, M. H. **Comparação de métodos de estimação para problemas com colinearidade e/ou alta dimensionalidade (p >n).** 2016. 65 f. Dissertação (Mestrado em Estatística) – Universidade Federal de São Carlos, São Carlos, 2016.

EFRON, B.; HASTIE, T.; JOHNSTONE, I.; TIBSHIRANI, R. Least Angle Regression. **The Annals of Statistics**, v. 32, n. 2, p. 407–451, 2003.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. **The Elements of Statistical Learning:** *data mining, inference and prediction.* Stanford: Springer, 2008.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. **Journal of Statistical Software**, v. 33, n. 1, p. 1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

HASTIE, T.; TIBSHIRANI, R.; TIBSHIRANI, R. **Extended Comparisons of Best Subset Selection, Forward Stepwise Selection and the Lasso** . *ArXiv e-prints*, , n. arXiv:1707.08692, 2017.

HASTIE, T.; TIBSHIRANI, R.; WAINWRIGHT, M. **Statistical learning with sparsity:** *The lasso and generalizations*. Seattle: Chapman and Hall Book, 2015.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning:** *with applications in R.* Nova York: Springer-Verlag, 2013.

MONTGOMERY, D. C.; PECK, E.A.; VINING, G. G. **Introduction to linear regression analysis.** Nova Jersey: Wiley, 5. ed., 2012.

MURPHY, K. P. **Machine Learning:** *a probabilistic perspective*. Londres: The MIT Press, 2012.

PEREIRA, L. S. **Geometria dos métodos de regressão lars, lasso e elastic net com uma aplicação em seleção genômica.** 2017. 167 f. Tese (Doutorado em Estatística e Experimentação Agropecuária) – Universidade Federal de Lavras, Lavras, 2017.

R Core Team. ***R: A Language and Environment for Statistical Computing***. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

TIBSHIRANI, R. Regression Shrinkage and Selection via the LASSO. **Royal**

Statistics Society, v. 58, n. 1, p. 267–288, 1996.

ZOU, H.; HASTIE, H.; TIBSHIRANI, R. On the degrees of freedom of the lasso. **The Annals of Statistics**, v. 35, n. 5, p. 2173–2192, 2007.