UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCELO BRUNO DE ALMEIDA VERAS

UMA VARIANTE DO MÉTODO DE REGRESSÃO FORWARD STAGEWISE PARA DADOS INCOMPLETOS

FORTALEZA

2017

MARCELO BRUNO DE ALMEIDA VERAS

UMA VARIANTE DO MÉTODO DE REGRESSÃO FORWARD STAGEWISE PARA DADOS INCOMPLETOS

Dissertação apresentada ao Curso de Programa de Pós-Graduação em Ciência da Computação do Departamento de Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. João Fernando Lima Alcântara

Co-Orientador: Prof. Dr. João Paulo Pordeus Gomes

FORTALEZA

2017

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)

V584v Veras, Marcelo.
Uma variante do método de regressão Forward Stagewise para dados incompletos / Marcelo Veras. – 2017.
54 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2017.
Orientação: Prof. Dr. João Fernando de Lima Alcântara.
Coorientação: Prof. Dr. João Paulo Pordeus Gomes.

1. Aprendizado de Máquina. 2. Regressão Linear. 3. Dados Faltantes. 4. Forward Stagewise. I. Título.
CDD 005

MARCELO BRUNO DE ALMEIDA VERAS

UMA VARIANTE DO MÉTODO DE REGRESSÃO FORWARDSTAGEWISE PARA DADOS INCOMPLETOS.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, da Universidade Federal do Ceará, como requisito para a obtenção do título de Mestre em Ciência da Computação.

Aprovada em 04 de abril de 2017.

BANCA EXAMINADORA

_____

Prof. Dr. João Fernando Lima Alcântara (Orientador)
Universidade Federal do Ceará – UFC

_____

Prof. Dr. João Paulo Pordeus Gomes (Coorientador)
Universidade Federal do Ceará – UFC

_____

Prof. Dr. Ajalmar Rêgo da Rocha Neto
Instituto Federal de Educação, Ciência
e Tecnologia do Ceará - IFCE

_____

Prof. Dr. Yuri Lenon Barbosa Nogueira
Universidade Federal do Ceará – UFC

Dedico essa dissertação a meu Pai William e minha mãe Edineusa que sempre lutaram pelos meus estudos, minha felicidade e minha saúde.

AGRADECIMENTOS

Á mulher da minha vida Lorena pelo seu apoio incondicional e sua resiliência nos momentos adversos.

Ao meu irmão William por sempre ser prestativo.

Aos Professores Dr. João Paulo Pordeus Gomes e Dr. João Fernando de Lima Alcântara e pelas orientações dadas no período do mestrado, com qual o trabalho tornou-se possível de ser realizado.

Aos Professores Antônio José Melo Leite Júnior e Dr. José Marcos Sasaki por terem me deram oportunidade e aconselhamento anteriormente ao período do mestrado, me possibilitando chegar onde cheguei.

Aos alunos do MDCC, meus colegas e amigos Jônatas Aquino, Diego Parente, Weslley Lioba, Alisson Alencar, Saulo Oliveira e Julio Sibaja pelas diversas discussões e aprendizados.

À toda a equipe do MDCC pela boa vontade e sempre estarem disponíveis para qualquer dúvida e solicitação.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela concessão da bolsa durante todo o período de realização deste mestrado.

Por fim, à todos que me apoiaram de alguma forma durante o caminho.

“No fim das contas, uma pessoa ama os próprios desejos e não aquilo que é desejado.”

(Friedrich Nietzsche)

RESUMO

O método de regressão Forward Stagewise é um popular algoritmo de regressão para conseguir modelos esparsos, ou seja, um modelo onde alguns coeficientes de regressão são nulos. Porém, em sua formulação somente são considerados os bancos de dados totalmente observáveis. Como em muitos casos os dados disponíveis nem sempre seguem tal hipótese, é necessário que eles passem por algum tipo de tratamento em uma etapa de pré-processamento ou deve-se utilizar um método produzido com o intuito de considerar tal característica. Neste trabalho realizamos uma extensão da regressão Forward Stagewise para tratar os casos em que os dados faltantes se fazem presentes na entrada do método. Tal extensão é obtida incutindo no método o procedimento para tratar os registros que possuem dados faltantes de forma que seja considerada a incerteza daqueles dados. Além disso, neste trabalho, realizamos um conjunto de testes sobre bancos de dados do mundo real, no qual obtivemos resultados positivos.

Palavras-chave: Aprendizado de máquina. Dados Faltantes. Regressão Linear. Modelos Esparsos. Regressão Forward Stagewise.

ABSTRACT

The Forward Stagewise regression is a popular regression method to achieve sparse models, that is, a model where some regression coefficients are null. However, in its formulation only fully observable databases are considered. As in many cases, available data do not always follow such hypothesis, and it is necessary that they pass through some kind of treatment in a pre-processing step or a method produced for the purpose of a particular characteristic must be used. In this work we extend the Forward Stagewise regression to treat cases where missing data is present at the input of the method. Such extension is obtained by including in the method the procedure for treating the records which have missing data so that the uncertainty of those data is considered. Beyond In this work, we performed a set of tests on the real world's databases in which we obtained positive results.

Keywords: Machine Learning. Missing Data. Linear Regression. Sparsity. Forward Stagewise Regression.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
ECM	Expectation Conditional Maximization
EM	Expectation Maximization
EQM	Erro Quadrático Médio
fdp	função de densidade de probabilidade
i.i.d	independentes e identicamente distribuidas
IM	Imputação Múltipla
IMC	Imputação por Média Condicional
LD	Listwise Deletion
MAR	*Missing at Random*
MCAR	*Missing Completely at Random*
MDQC	Média da Diferença Quadrática entre os Coeficientes
MLM	Minimal Learning Machine
MNAR	*Missing Not at Random*
PD	Pairwise Deletion
RFS	Regressão Forward Stagewise
RFSDF	Regressão Forward Stagewise para Dados Faltantes
SFS	Seleção Forward Stepwise

LISTA DE SÍMBOLOS

X	Matriz de entrada
Y	Vetor de rótulos
θ	Vetor de coeficientes de regressão
$\mathbf{x}_i$	Vetor com os elementos da linha i da matriz X
X_j	Vetor com os elementos da coluna j da matriz X
$x_{i,j}$	elemento da linha i e coluna j de X
y_i	elemento i de Y
θ_j	Elementos j de θ
M_i	vetor de índices dos elementos para qual $\mathbf{x}_i$ é não-observável
O_i	vetor de índices dos elementos para qual $\mathbf{x}_i$ é observável
$\mathbf{x}_{i,M}$	Elementos não observáveis de $\mathbf{x}_i$
$\mathbf{x}_{i,O}$	Elementos observáveis de $\mathbf{x}_i$
μ	Vetor com as médias das colunas de X
Σ	Matriz da covariância de X

SUMÁRIO

1 INTRODUÇÃO

O Aprendizado de Máquina (AM) é uma área da Inteligência Artificial que usa algoritmos para analisar dados, aprender com eles e, em seguida, fazer uma determinação ou previsão sobre algo no mundo.

Segundo Mitchell (1997, p. 2,tradução livre do autor):

> "Um computador aprende de uma experiência "E"' em respeito a uma tarefa "T" e uma performance medida por "P", se sua performance na tarefa T, medida pela performance P, melhora com experiência E."

Nesse contexto, podemos dizer que a experiência é contabilizada a partir dos dados que são apresentados ao programa. De tais dados, são extraídos propriedades e relacionamentos. O AM é dividido principalmente em dois paradigmas de aprendizagem: o supervisionado e o não-supervisionado.

O aprendizado supervisionado é a tarefa que infere uma função a partir de dados de treinamento rotulados. Os dados de treinamento são representados por um conjunto de exemplos. Cada exemplo tem um rótulo que corresponde à saída esperada daquele exemplo. Analisando cada exemplo, o algoritmo pode ajustar a função inferida baseada em seu rótulo. Assim, o aprendizado pode ser genérico o suficiente para poder realizar previsões sobre dados futuros.

O aprendizado não supervisionado abrange os métodos de treinamento sem rotulação em suas entradas. Em tais entradas, deve-se encontrar algum padrão ou relação oculta em suas estruturas. Uma de suas tarefas mais significativas é utilizar as entradas para a criação de diferentes agrupamentos e classificar a qual destes agrupamentos uma nova entrada viria a pertencer. Já que os exemplos não possuem uma saída esperada, não faz sentido computar a precisão da estrutura gerada.

Em anos recentes, universidades, institutos de pesquisa e empresas têm aplicado métodos de aprendizado de máquina em diversos problemas, desde sistemas de recomendação de compras online (LINDEN *et al.*, 2003) até segurança contra fraudes em transações financeira (BOLTON; HAND, 2002). Apesar da grande popularidade desses métodos, a formulação original dos mesmos não é capaz de lidar com algumas anomalias que aparecem em conjuntos de dados reais. Dentre estas anomalias podemos destacar a presença de dados faltantes.

Dados faltantes ocorrem quando valores de um ou mais registros de um banco de dados, por qualquer motivo, não existem ou são inacessíveis. Isso pode ocorrer devido a uma

grande quantidade de fatores, tais como: corrompimento no arquivo do banco de dados, falha nos equipamentos de medição, não-preenchimento ou retirada deliberada dos dados, etc. (LITTLE; RUBIN, 2002). Usualmente, tratamos este problema em uma etapa de pré-processamento, onde os dados faltantes são descartados ou preenchidos com valores prováveis. Os mais diversos métodos podem ser utilizados para completar os dados: eles podem ser tão simples quanto a substituição pela média, ou complicados como em Abdella e Marwala (2005), onde é utilizado um algoritmo genético para minimizar uma função de erro derivada de uma rede neural.

Recentemente, variações dos métodos de AM tem sido propostas. Em tais variações, o pré-processamento não é necessário e o tratamento dos dados faltantes é realizado de forma intrínseca ao método. Em Eirola *et al.* (2013) foi proposto o método onde se calcula o valor esperado da distância entre dois vetores com dados faltantes, ao invés de preencher cada valor individualmente e calcular a distancia posteriormente. Motivado por este resultado, Mesquita *et al.* (2015) realizou uma aplicação da técnica no método Minimal Learning Machine (MLM) (JUNIOR *et al.*, 2013), para criar uma versão robusta a dados faltantes. Já em Belanche *et al.* (2014) foi apresentado uma forma de estender funções de kernel para lidar com dados faltantes, novamente sem realizar o pré-processamento das entradas. Estes métodos apresentaram resultados promissores além de resultarem em técnicas mais elegantes para o tratamento de dados faltantes.

Inspirados por estes trabalhos propomos uma variação do método de Regressão Forward Stagewise RFS (EFRON *et al.*, 2004). A regressão Forward Stagewise é um método iterativo de regressão linear que pode ser utilizado para obter modelos esparsos. Assim como diversos outros métodos de aprendizagem de máquina, a proposta original do RFS assume que os dados de entrada são sempre completos.

1.1 Objetivo Geral

Objetiva-se neste estudo construir um método que realize uma regressão esparsa baseada na RFS, além disso o método deve ter o recurso necessário para lidar com a possibilidade da existência de dados faltantes nos dados de entrada levando em conta a incerteza inerente destes.

1.2 Objetivo Específico

O método deve ser capaz de gerar a mesma saída que a RFS para bancos de dados que não possuem dados faltantes. Deve ser capaz de gerar modelos lineares esparsos para qualquer base que possua a característica de dados faltantes. Além de que, deve produzir modelos mais assertivos que as técnicas de pré-processamento caso os dados estejam dispostos nas premissas deste trabalho.

1.3 Contribuição do Autor

Este trabalho apresenta um novo algoritmo de regressão linear robusta a dados faltantes. Tal método fora nomeado Regressão Forward Stagewise para Dados Faltantes (RFSDF), pois o mesmo tem como base o método RFS. Sua principal diferença ante o RFS é poder computar de maneira robusta dados incompletos sem antes passar por etapas de pré-processamento. Também é apresentado uma série de comparações em 5 bancos de dados do mundo real com métodos de tratamentos diferentes. Para cada método, variamos a quantidade de dados faltantes, assim analisando como essas condições afetam os modelos gerados.

1.4 Publicações

Relativo ao desenvolvimento deste trabalho a seguinte publicação foi realizada:

Marcelo B. A. Veras; Diego P. P. Mesquita; João P. P. Gomes; Amauri H. Souza Junior e Guilherme A. Barreto, **Forward Stagewise Regression on Incomplete datasets.** 2017 Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks, 386-395

Simultaneamente, o autor realizou contribuições para a seguinte publicação:

Francisco F. R. Damasceno; Marcelo B. A. Veras; Diego P. P. Mesquita; João P. P. Gomes e Carlos E. F. d. Brito, **Shrinkage k-Means: A Clustering Algorithm Based on the James-Stein Estimator.** 2016 5th Brazilian Conference on Intelligent Systems (BRACIS), 433-437

1.5 Estrutura da Dissertação

O restante deste trabalho se divide em 4 capítulos. O capítulo 2 abrange uma revisão dos principais conceitos abordados para a realização do método proposto. No capítulo 3, exibiremos a elaboração do método proposto e destacamos os procedimentos auxiliares utilizados em sua construção. Os resultados obtidos pelos método são analisados e comparados no capítulo 4. Por último, no capítulo 5, recapitulamos o problema exposto e a solução apresentada. Nesse último capítulo também serão discutidos os trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS

Na primeira seção deste capítulo, exibiremos o modelo linear para regressão com sua formulação e treinamento. Nesta seção também serão detalhados métodos de aprendizagem que resultam em modelos lineares esparsos e o algoritmo RFS. Na segunda parte abordaremos dados incompletos, apresentando seus mecanismos e formas de tratamentos tradicionalmente utilizadas.

2.1 Regressão Linear

Regressão linear é a forma mais utilizada para analisar os dados e obter predições de uma variável dependente através de um conjunto de variáveis independentes. Neste trabalho utilizaremos a notação $\mathbf{x}$ para denotar o vetor que contém as variáveis independentes e y para denotar a variável dependente de $\mathbf{x}$. $\mathbf{x}$ é também chamado de vetor de características e é um vetor multidimensional em $\mathbb{R}^p$ tal que $\mathbf{x}^\intercal = [x_1, x_2, \cdots, x_p]$. y é também conhecido como rótulo ou variável objetivo, e está em $\mathbb{R}$. Um par $(\mathbf{x}, y)$ é um exemplo de treinamento e a base de dados que utilizamos para realizar o aprendizado é composta por n exemplos. X é a matriz que contem todos os n vetores de características, tal que $\mathbf{x}_i$ é o vetor com os valores correspondentes a i-ésima linha de X e $i \in \{1, \cdots, n\}$. Da mesma forma Y é vetor de variáveis dependentes, tal que y_i é o i-ésimo elemento de Y e $i \in \{1, \cdots, n\}$.

No aprendizado supervisionado o objetivo é achar uma função f tal que $f(\mathbf{x})$ é um bom preditor de y. Para isso, precisamos escolher uma representação para essa função. A solução adotada consiste em um modelo linear da seguinte forma:

$$f_\theta(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_p x_p \tag{2.1}$$

Os θ_j são os coeficientes de regressão (também chamados de pesos) e mapeiam o espaço de $\mathbf{x}$ no espaço de y. Para simplificar a notação podemos atribuir a x_0 o valor de 1 e multiplica-la por θ_0 para obtermos a seguinte equação:

$$f_\theta(\mathbf{x}) = \sum_{j=0}^{p} \theta_j x_j = \mathbf{x}^\intercal \theta \tag{2.2}$$

Dado uma conjunto de dados de treinamento, um problema que aparece é: qual conjunto de θ_j's devemos selecionar para que cada $f(\mathbf{x}_i)$ seja próximo de seu rótulo y_i correspon-

dente? Assim, podemos definir a função de custo em relação à soma dos resíduos ao quadrado, onde o resíduo é dado pela diferença entre $f_\theta(\mathbf{x})$ e a saída objetivo y:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(f_\theta(\mathbf{x}_i) - y_i)^2 \tag{2.3}$$

e objetivamos escolher um θ que minimize tal função.

O algoritmo do gradiente descendente acha o θ que minimiza[1] a função de custo; para isso, ele atualiza todos os pesos simultaneamente em uma fração α contrária a do gradiente, de tal maneira que:

$$\theta_j \leftarrow \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta) \tag{2.4}$$

em que α representa a taxa de aprendizado.

Dado que $\frac{\partial}{\partial\theta_j}f_\theta(\mathbf{x}_i) = \mathbf{x}_i$, podemos desenvolver a derivada da equação 2.4 da seguinte maneira:

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}\sum_{i=1}^{n}(f_\theta(\mathbf{x}_i) - y_i)^2 \\
&= \sum_{i=1}^{n}\frac{1}{2}\frac{\partial}{\partial\theta_j}(f_\theta(\mathbf{x}_i) - y_i)^2 \\
&= \sum_{i=1}^{n}\frac{1}{2}\frac{\partial}{\partial\theta_j}(f_\theta(\mathbf{x}_i)^2 - 2f_\theta(\mathbf{x}_i)y_i + y_i^2) \\
&= \sum_{i=1}^{n}\frac{1}{2}(2\mathbf{x}_i f_\theta(\mathbf{x}_i) - 2\mathbf{x}_i y_i) \\
&= \sum_{i=1}^{n}\mathbf{x}_i(y_i - f_\theta(\mathbf{x}_i))
\end{aligned} \tag{2.5}$$

Obtendo assim a regra de atualização dos pesos para cada amostra, da seguinte maneira:

$$\theta_j \leftarrow \theta_j + \alpha(f_\theta(\mathbf{x}_i) - y_i)x_{i,j} \tag{2.6}$$

Para realizar a atualização dos pesos em relação a todo o banco de dados, executamos um loop em todos os n exemplos até que o vetor θ convirja.

A maneira anterior é uma forma iterativa de encontrar θ, porém existe uma maneira onde se utiliza a totalidade da matriz de treinamento em um único passo, e para derivar tal método iremos rearranjar a função de custo inicial, de modo que, todos os membros iram aparecer de forma vetorial ou matricial. Dessa maneira podemos obter a seguinte equivalência:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(f_\theta(\mathbf{x}_i) - y_i)^2 \equiv \frac{1}{2}(X\theta - Y)^\intercal(X\theta - Y) \tag{2.7}$$

[1] Para o gradiente descendente minimizar a função, é necessário que alguns critérios sejam alcançados, tal como: a função ter um mínimo, ser diferenciável, utilizar um passo de aprendizagem α não muito grande e não ter uma regra de convergência muito rigorosa.

e derivando $J(\theta)$ em relação a θ obtemos:

$$\begin{aligned}\frac{\partial}{\partial\theta}J(\theta) &= \frac{\partial}{\partial\theta}\frac{1}{2}(X\theta - Y)^\intercal(X\theta - Y)\\ &= \frac{1}{2}\frac{\partial}{\partial\theta}((X\theta)^\intercal(X\theta) - ((X\theta)^\intercal Y) - (Y^\intercal(X\theta)) + (Y^\intercal Y))\\ &= \frac{1}{2}(((X^\intercal X + X^\intercal X)\theta) - 2(\theta^\intercal X^\intercal Y))\\ &= X^\intercal X\theta - X^\intercal Y\end{aligned} \tag{2.8}$$

por fim, obtemos θ que minimiza a função de custo igualando a equação 2.8 a zero:

$$\begin{aligned}X^\intercal X\theta - X^\intercal Y &= 0\\ X^\intercal X\theta &= X^\intercal Y\\ \theta &= (X^\intercal X)^{-1}X^\intercal Y\end{aligned} \tag{2.9}$$

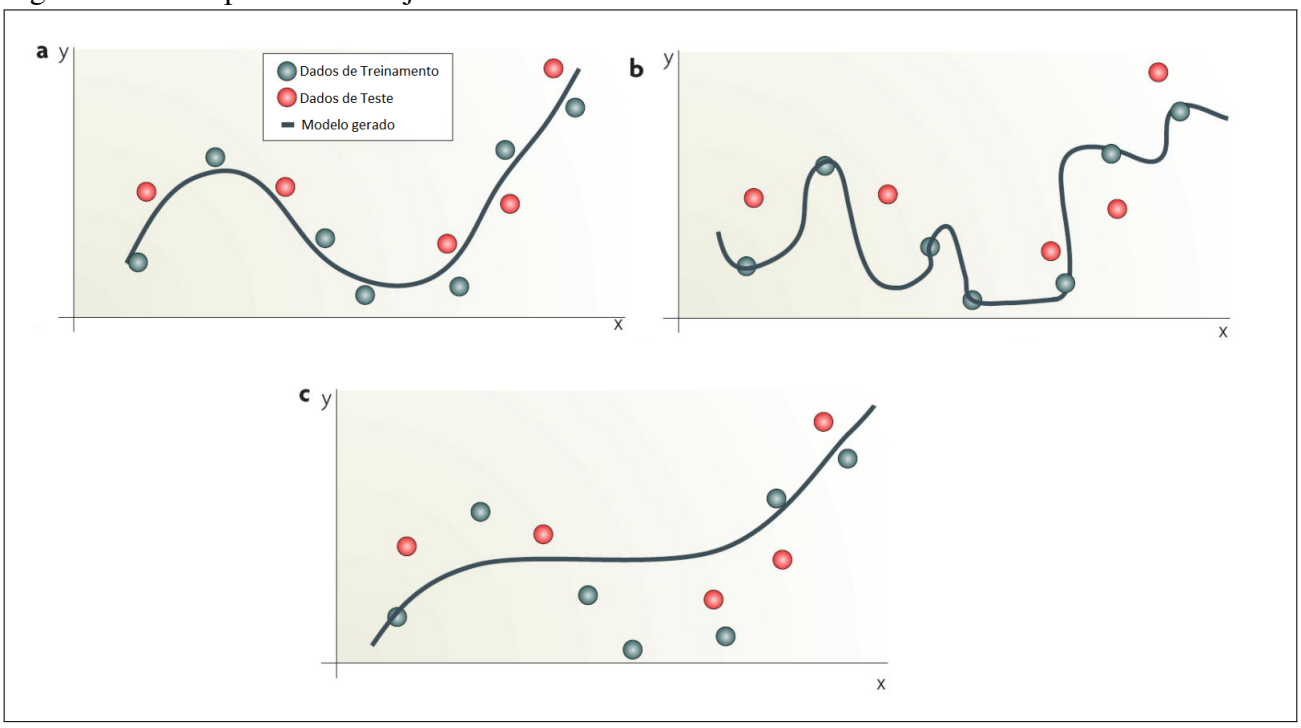
. A equação 2.9 é uma fórmula fechada para obtenção do θ que minimiza a função de custo $J(\theta)$.

No mundo real ao tentar construir nosso modelo alguns problemas podem aparecer. Por exemplo: Tentando estimar o valor de um imóvel, um corretor realiza uma regressão linear e define como variável independente a quantidade de quartos. Obviamente logo ele percebe que aquele modelo é muito simples: tal modelo não prediz novas entradas corretamente. Ele então adiciona novas variáveis como: tamanho, número de suítes, localização. O modelo passa a ter mais poder preditivo e ser mais complexo. Não satisfeito com os resultados obtidos, o corretor adiciona a quantidade de banheiros, de andares, de portas, de janelas e a média da idade da última família. Então ele percebe que seu modelo prevê muito bem os dados de treinamento, mas piora a predição de novas entradas. O seu modelo sofreu de sobre-ajuste e tem uma péssima generalização. Quando realizamos uma regressão, espera-se que o ruído do conjunto de treinamento não seja incorporado no modelo. Na imagem 1 podemos observar um modelo com ajuste próximo ao ideal em **a**, sofrendo de sobre-ajuste em **b** e com um ajuste abaixo do esperado em **c**.

Por ser uma técnica que limita a complexidade do modelo, a regularização se torna uma estratégia natural ante o sobre-ajuste. Nela, o vetor de pesos é limitado por algum fator, seja na norma ou na quantidade das características que irão compô-lo (BISHOP, 2006).

No método dos mínimos quadrados a regularização se faz na função de custo $J(\theta)$. Ela é subordinada a algum limite λ na norma ℓ_2 de θ, sendo um compromisso entre o quão bem

Figura 1 – Exemplo de sobre-ajuste dos dados



Fonte: (CLARKE *et al.*, 2008, Modificado pelo Autor)

o método se ajusta aos dados e o quão grande a norma de pesos pode ficar.

$$J(\theta) = \frac{1}{2}(X\theta - Y)^\intercal(X\theta - Y) + \frac{1}{2}\lambda\theta^\intercal\theta \tag{2.10}$$

De forma semelhante a equação 2.8 e 2.9 podemos realizar a derivada da função de custo 2.10 em relação a θ seguinte maneira:

$$\begin{aligned}
\frac{\partial}{\partial\theta}J(\theta) &= \frac{\partial}{\partial\theta}\left(\frac{1}{2}(X\theta - Y)^\intercal(X\theta - Y) + \frac{1}{2}\lambda\theta^\intercal\theta\right) \\
&= \frac{1}{2}\frac{\partial}{\partial\theta}((X\theta)^\intercal(X\theta) - ((X\theta)^\intercal Y) - (Y^\intercal(X\theta)) + (Y^\intercal Y) + \lambda(\theta^\intercal\theta)) \\
&= \frac{1}{2}(((X^\intercal X + X^\intercal X)\theta) - 2(\theta^\intercal X^\intercal Y) + \lambda 2I\theta) \\
&= X^\intercal X\theta - X^\intercal Y + \lambda I\theta
\end{aligned} \tag{2.11}$$

e novamente igualamos a zero, obtendo:

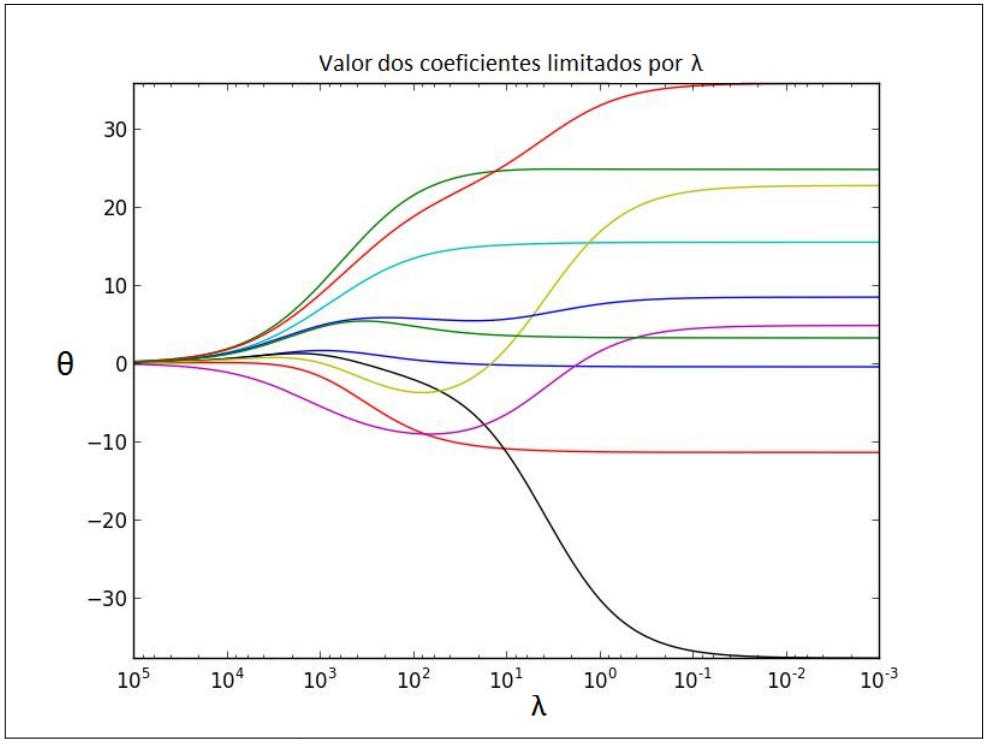
$$\begin{aligned}
X^\intercal X\theta - X^\intercal Y + \lambda I\theta &= 0 \\
X^\intercal X\theta + \lambda I\theta &= X^\intercal Y \\
(X^\intercal X + \lambda I)\theta &= X^\intercal Y \\
\theta &= (X^\intercal X + \lambda I)^{-1}X^\intercal Y
\end{aligned} \tag{2.12}$$

2.1.1 Modelos lineares esparsos

A regularização apresentada na seção anterior é também nomeada de regularização Tikhonov, pois foi nomeada em homenagem a Andrey Tikhonov. Nela, o parâmetro λ é utilizado para restringir a norma ℓ_2 dos pesos. Assim os coeficientes de regressão não atingem números grandes em valores absolutos.

Em uma análise menos detalhada do valor de λ, podemos rapidamente observar que λ's próximos a 0 nós retornará aproximadamente a mesma solução dos mínimos quadrados padrão. Ao passo que, se aumentarmos tal parâmetro, obtemos uma solução que comprime até que, para um valor muito alto, todos os coeficientes tenderão a um valor nulo, como podemos observar na figura 2.

Figura 2 – Regularização Tikhonov



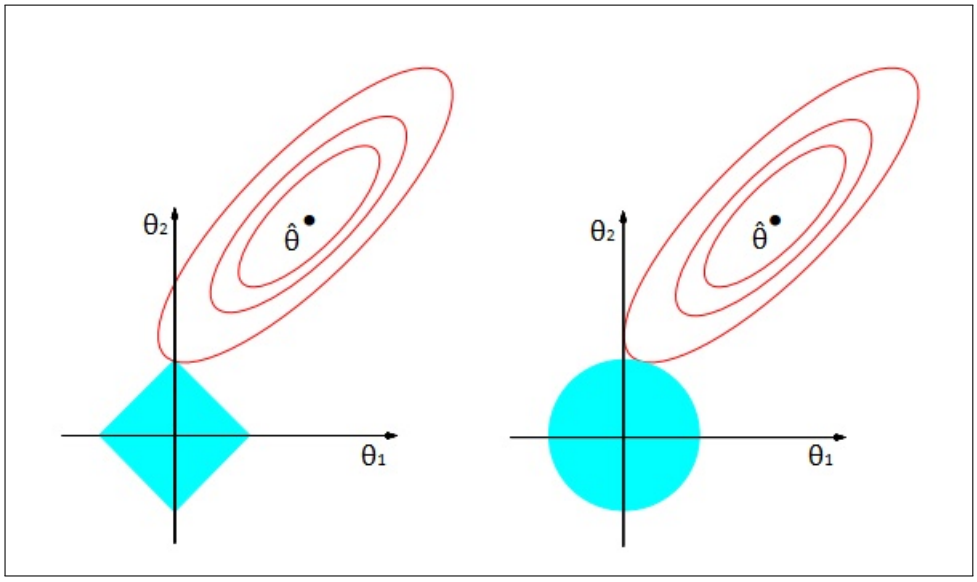
Fonte: Autor desconhecido[1]

Porém alguns problemas podem demandar uma solução esparsa, em que a maioria dos coeficientes seja zero e como podemos constatar a regularização Tikhonov não possui tal propriedade. Segundo Hastie *et al.* (2015, p. 7), diminuir a variância e aumentar o viés dos resultados do modelo restringindo alguns coeficientes ou mesmo zerando-os, pode tornar as predições mais acurada e ajudar a identificar quais os coeficientes tem maiores efeitos sobre nosso resultado.

Em Tibshirani (1994) é proposto o método lasso (*least absolute shrinkage and selection operator*) que pode retornar resultados esparsos. Essa "esparsidade" no resultado é gerada pela natureza da solução dada pelo método, no qual utiliza a função de custo $J(\theta) = \frac{1}{2}(X\theta - Y)^\intercal(X\theta - Y) + \lambda||\theta||_1$.

A figura 3 exemplifica de forma simplificada como o método lasso (à esquerda) resulta em uma solução esparsa, se comparado a regressão com regularização ℓ_2 (à direita). Por se tratar de um exemplo para fácil entendimento visual, nos gráficos só são apresentados 2 coeficientes de regressão, θ_1 e θ_2. Porém podemos facilmente estende-lo para um espaço p-dimensional, com p coeficientes. Sendo uma valoração para o conjunto $\theta = (\theta_1, \cdots, \theta_p)$ representada por um ponto no gráfico. Então definimos $\hat{\theta}$ como o modelo linear obtido pela versão sem regularização dos mínimos quadrados para θ (equação 2.9) (HASTIE *et al.*, 2015). Cada elipse que circunda $\hat{\theta}$ são soluções de θ para qual a soma dos resíduos ao quadrado é a mesma. Quanto menor a distância delas de $\hat{\theta}$ mais perto da solução linear ótima. A área representada em azul são as regiões onde se encontram o conjunto de resultados gerados através da restrição em θ gerada pelas regras $\sum_{j=1}^{p} |\theta_j| \leq \lambda$ e $\sum_{j=1}^{p} \theta_j^2 \leq \lambda^2$, respectivamente. Na regressão lasso, a figura geométrica gerada pelo modelo tem maior probabilidade de encontrar uma solução onde alguns coeficientes sejam nulos. Já na regularização Tikhonov, mesmo que os coeficientes sejam pequenos ele normalmente não são nulos.

Figura 3 – Coeficientes com regularização ℓ_1 a esquerda e ℓ_2 a direita



Fonte: (HASTIE *et al.*, 2015, p. 11)

[1] <https://statcatinthehat.wordpress.com/2014/07/16/regularized-regression-ridge-in-python-part-2/> Acessado em 13/01/2017

2.1.2 Regressão Forward Stagewise

Outra forma de se obter um modelo de regressão esparsa é através dos algoritmos de Seleção *Forward* ou Seleção Forward Stepwise (SFS). A definição do método SFS apresentada em Efron *et al.* (2004) se baseia em Weisberg (1980) e diz: dado um conjunto de possíveis preditores X_j, sendo X_j a j-ésima coluna de X, normalizados com média zero e variância um, escolha o preditor que tenha maior correlação absoluta com a saída Y, por exemplo X_1, e realize uma simples regressão linear de Y nessa característica. Ao reduzir Y do atual modelo o resíduo restante é ortogonal à X_1, este resíduo agora é considerado a saída-alvo. Nós projetamos as outras colunas ortogonalmente à X_1 e repetimos o processo de seleção. Em k passos temos, um grupo de preditores X_1, X_2, ..., X_k, sendo k menor que o número de colunas em X, e o restante dos coeficientes do modelo será nulo. A estratégia gulosa proposta pelo SFS muitas vezes deixa alguns bons preditores de fora por estarem altamente correlacionados com algum $X_l \in X_1$, X_2, ..., X_k. O método RFS, que é uma versão mais ponderada do SFS, constrói uma solução iterativamente a partir de centenas ou milhares de pequenos passos como podemos ver no algoritmo 1:

Algoritmo 1: Regressão Foward Stagewise

início

Normalize as colunas de X_J e centralize Y

$\mathbf{r} = Y$;

$\theta_1, \theta_2, ..., \theta_p = 0$

enquanto *Não convergir* **faça**

ache o preditor X_j mais correlacionado com r

atualize $\theta_j = \theta_j + \delta$, onde $\delta = \varepsilon$ sinal$(< \mathbf{r}, X_j >)$,

sendo ε o passo que cada θ dará em direção a maior correlação $(0 < \varepsilon \ll 1)$

$\mathbf{r} = \mathbf{r} - \delta X_j$

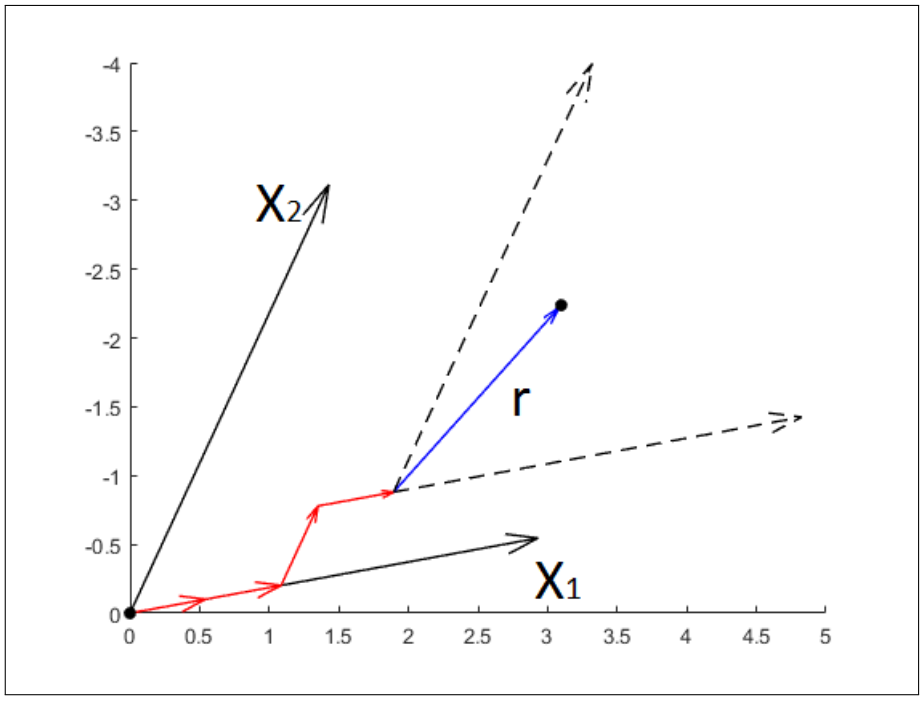
fim

fim

Ou seja, o algoritmo começa com um modelo nulo e resíduo r inicial igual a Y, então identifica qual preditor é o mais correlacionado com r, digamos X_1, e adiciona um pequeno incremento em θ_1 com mesmo sinal da correlação entre X_1 e r, repete esse procedimento até que exista algum X_2 que seja tão ou mais correlacionado quanto X_1 e realiza o mesmo procedimento para ele. Essa operação é repetida até que sempre o mesmo preditor seja atualizado para valores

obtidos em iterações anteriores (critério de convergência). Uma execução para dois preditores pode ser vista na Figura 4, nela as setas vermelhas indicam as iterações já realizadas, no cenário representado foram realizadas 4 iterações, em azul podemos ver o resíduo após resultante do modelos obtido após a quarta iteração; ainda podemos observar que: na iteração 1, 2 e 4 X_1 era o atributo mais correlacionado com o resíduo, enquanto na terceira iteração X_2 passou a ser o mais correlacionado. Em caso de interrupção prematura do algoritmo podemos constatar que um resultado esparso será obtido como podemos perceber na Figura 5.

Figura 4 – Exemplo de execução do algoritmo RFS



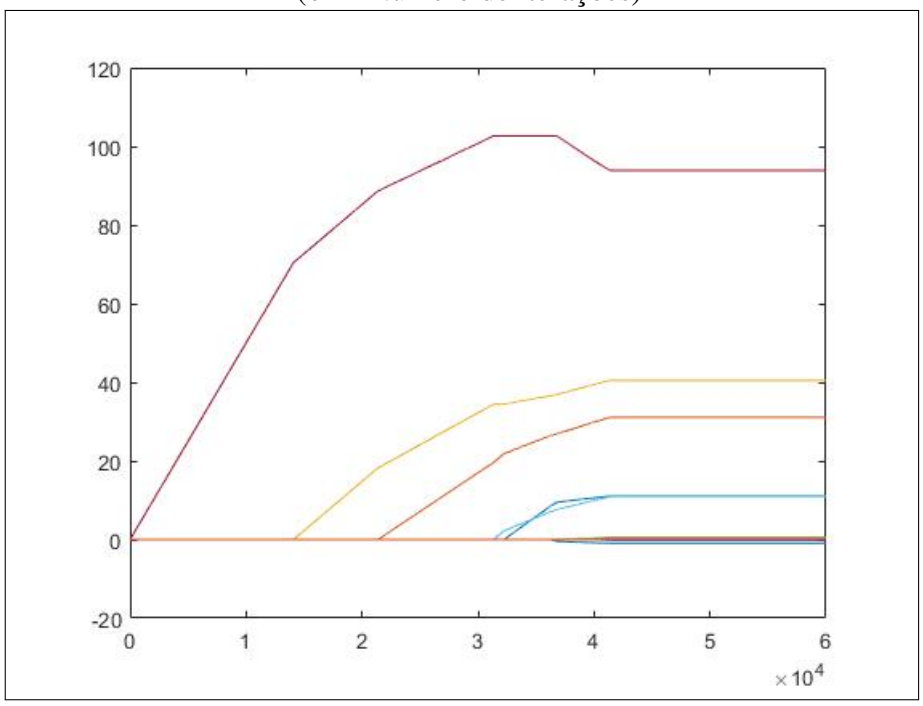
Fonte: Elaborada pelo autor

O RFS assume que os dados apresentados serão sempre observáveis (uma premissa que diversas vezes pode ser falha). Nestes casos, etapas de pré-processamento devem ser realizadas nos dados, porém neste trabalho é apresentado um método com base no próprio RFS que é robusto a estas situações.

2.2 Dados Faltantes

Os algoritmos de aprendizado de máquina ajustam os modelos criados aos dados de treinamento. Portanto qualquer anomalia nas informações fornecidas podem ocasionar uma perda no poder de dedução do modelo final. Dentre os tipos de anomalias existentes, uma especialmente lesiva são os dados faltantes. Diversas áreas de pesquisa possui problemas com

Figura 5 – Exemplo de execução do algoritmo RFS
(θ $\times$ Número de iterações)



Fonte: Elaborada pelo autor

alta incidência de dados faltantes em bancos de dados do mundo real e pode dificultar deveras a análise de dados (DONG; PENG, 2013). Diversos motivos podem fazer com que os dados estejam omissos, tais como: corrompimento dos dados, falhas na medição, não preenchimento ou mesmo a retirada proposital. Damos o nome de dados faltantes a todos os casos que, por qualquer motivo, tenham dados indisponíveis.

Por possuir diversas causas é importante salientar que cada instância de problema de dados faltantes pode possuir propriedades específicas e para cada uma delas existem métodos indicados para trata-los. Little e Rubin (2002) classifica os dados faltantes em 3 tipos de mecanismos diferentes: *Missing Completely at Random* (MCAR), *Missing at Random* (MAR) e *Missing Not at Random* (MNAR). Cada mecanismo relaciona a probabilidade de um dado estar oculto com valor dos dados que são observáveis e faltantes. Considerar estes mecanismos podem ajudar na escolha de técnicas apropriadas para minimizar o impacto que os dados faltantes causam na regressão.

Na classificação de Little e Rubin: MCAR indica que os dados são faltantes independentemente do valor das variáveis observáveis e não-observáveis. Dado que X é uma matriz de entrada, R é uma matriz onde o elemento de índice (i, j) tem valor 1, se o elemento $x_{i,j}$ da matriz X for faltoso e 0 caso contrário. Então MCAR pode ser representado em termos matemáticos por: $P(R|X_{obs}, X_{mis}) = P(R)$. No mecanismo MAR os dados faltosos dependem

dos dados observáveis, porem continuam sem depender da própria valoração e do valor de outras variáveis não observadas, matematicamente temos: $P(R|X_{obs}, X_{mis}) = P(R|X_{obs})$. A terceira classificação possível, MNAR, é dada quando o padrão de perda da informação depende tanto do valor dos atributos faltosos quanto dos observáveis. Esse padrão é o mais difícil de ser tratado, pois qualquer técnica utilizadas para realizar inferência sobre os dados completos não pode ignorar os dados faltantes (AMBROSIUS, 2007).

Para melhor elucidar cada tipo de mecanismo, podemos elaborar um situação hipotética de suas ocorrências: em um formulário, no qual, pessoas são entrevistadas sobre sua saúde, são separados três grupos de participantes para responde-lo, cada grupo decide se responde a pergunta sobre depressão de acordo com um determinado mecanismo. O primeiro grupo de participantes joga uma moeda, a face da moeda define se a pergunta é respondida ou não. Neste caso responder o item independe de qualquer outra resposta do formulário, mesmo da resposta sobre depressão, logo o mecanismo de falta é MCAR. No segundo grupo, as pessoas que responderam masculino na questão sobre sexo, tem maior tendencia a não responder sobre depressão, devido a homens terem preconceito com doenças mentais. Neste caso temos um atributo observável alterando diretamente a chance de responderem a outra questão, logo a probabilidade da questão possuir algum valor depende de algum valor observável, essa é a característica do MAR. Por último podemos ter a situação aonde as pessoas que possuem depressão são menos propensas a responderem a questão, pois sentem-se desconfortáveis. O fato do próprio valor da pergunta definir se ela será ou não respondida se encaixa em nosso último mecanismo MNAR, pois a chance do valor ser faltante aumenta caso a resposta da questão sobre depressão seja afirmativa.

2.2.1 Tratamento de dados faltantes

Na seção anterior foi abordado um pouco da natureza dos dados faltantes, podemos agora nos focar em como lidar com eles. Nesta seção é fornecido uma breve revisão sobre as principais linhas de tratamento utilizadas.

Dentre as formas de tratamento, as mais usuais são aquelas que utilizam apenas a parte dos dados que é observável (PEUGH; ENDERS, 2004). Isto é, dada uma base de dados que contem dados faltantes, ignora-se ou remove-se a parte contaminada desses dados e utiliza-se a restante. A principal diferença entre os métodos que utilizam esta estratégia é a forma como esta remoção é executada e dentre elas as mais notáveis são o Listwise Deletion (LD) e o Pairwise Deletion (PD). O algoritmo LD remove todos os registros que contiverem pelo menos um atributo

não preenchido, enquanto o PD não exclui todo o registro que contem dados faltantes, mas ignora as características faltantes na hora de calcular propriedades de algum conjunto de dados. Esses algoritmos supõem que os dados faltantes são MCAR, pois caso contrário o resultado será enviesado. Por exemplo, em nossa situação hipotética anteriormente apresentada, caso os formulários do segundo grupo fossem tratados com LD o modelo não irá condizer com a realidade pois não levará em conta uma parte dos participantes masculinos. De acordo com Acuna e Rodriguez (2004), taxas entre 1% e 5% de dados faltosos são consideradas triviais, porem mais do que 5% podem requerer processos mais sofisticados para trata-los.

A técnica de imputação simples dos dados é uma estratégia adicional à apresentada acima, nela as entradas faltosas são substituídas por valores e tratadas como se fossem variáveis observáveis. Ao nos desfazer de uma parte das entradas acabamos perdendo informação importante para construção de nosso modelo portanto, a imputação aparece como uma forma de tentar recuperar parte desta informação com o menor enviesamento possível. Existem diversas opções para escolha da maneira que os dados serão imputados, e algumas delas são triviais. A maneira mais fácil de realizar tal feito é pela média simples, porém, apesar de manter a média inalterada esta maneira pode distorcer de forma grave a distribuição das variável que sofrem imputação modificando algumas de suas medições, tal como, o desvio padrão. Uma opção é adicionar um campo extra indicando a omissão de alguns dados, isso juntamente a imputação por média simples pode aliviar um pouco o enviesamento provocado. Outra forma de realizar o procedimento é realizar uma regressão nos dados observáveis e partir disso utilizar o resultado para sobrepor os valores omissos. Neste caso a problemática é encontrar um bom modelo de regressão.

Outra forma bastante usual consiste na técnica denominada Imputação por Média Condicional (IMC). A IMC é uma técnica que preenche os componentes faltantes de um vetor, de acordo com a esperança do valor destes elementos condicionados aos componentes observáveis do mesmo vetor. Em geral, podemos assumir que os dados tem qualquer distribuição, sendo a distribuição normal multivariada a mais comum.

É importante salientar que a imputação é um processo de estimação, por conta disso, existe uma incerteza associada ao valores que serão imputados. Os métodos de imputação simples não contabilizam esta incerteza, pois utilizam estimativas pontuais. Com o intuito de resolver este problema foram criados os chamados métodos de Imputação Múltipla (IM)(RUBIN, 1987). Nesta abordagem são realizadas múltiplas imputações gerando assim múltiplos conjuntos

de dados. A geração de diferentes valores para imputação está associada as distribuições das estimativas dos valores faltantes. Estas distribuições são normalmente obtidas após a suposição de um modelo paramétrico para a distribuição dos dados. Tendo os diversos bancos de dados completos, são construídos vários modelos (um para cada conjunto de dados) e os resultados destes modelos são combinados (WAYMAN, 2003). Apesar de trazer bons resultados, IM pode ser computacionalmente bastante custoso devido a necessidade de treinamento de diversos modelos.

Conforme dito anteriormente, muitas técnicas de imputação necessitam de um modelo estatístico para a distribuição do conjunto de dados. Se esta distribuição não estiver especificada, podemos supor algum modelo paramétrico/não-paramétrico. Dado um modelo, faz-se então necessária a estimação dos parâmetros da distribuição. Diferentemente da estimação de parâmetros para o caso completo, a estimação dos parâmetros de uma distribuição a partir de dados com atributos faltantes não pode ser realizada através de estimativas de máxima verossimilhança de forma direta. Em conjunto de dados incompletos, os valores utilizados para os dados faltantes influenciam na estimação dos parâmetros da distribuição. Desta forma, pode-se então utilizar o algoritmo Expectation-Maximization

2.2.2 Expectation Maximization

Primeiramente introduzido em Dempster *et al.* (1977), o Expectation Maximization (EM) é um método iterativo para resolver problemas difíceis de máxima verossimilhança na presença de variáveis latentes. Variáveis latentes são variáveis que não podem ser diretamente observadas mas, tem seu valor inferido por outras variáveis observáveis. Em nosso contexto os dados faltantes de um registro são tratados como variáveis latentes.

Suponha um conjunto de n amostras aleatórias $\mathbf{x}_i$, para todo $i \in \{1, \cdots, n\}$, as quais são independentes e identicamente distribuidas (i.i.d). A variável aleatória da qual elas foram observadas tem uma função de densidade de probabilidade (fdp), definida por $f(\mathbf{x}|\beta)$ com $\beta \in \beta$, onde β é o espaço paramétrico. Cada vetor $\mathbf{x}_i$ poderá conter elementos faltosos e observáveis entre suas características. Iremos dizer que $x_{i,j}$ é faltante se o j-ésimo elemento elemento de $\mathbf{x}_i$ não é observável, então j pertence ao grupo dos elementos não observáveis M_i, caso contrário j pertencerá ao grupo O_i. Então indicaremos todos os elementos não observáveis por $\mathbf{x}_M$ e o restante por $\mathbf{x}_O$, logo $\mathbf{x}_i \sim (\mathbf{x}_M, \mathbf{x}_O)$. Estimar a máxima verossimilhança em um conjunto que tenha tais características é considerado uma tarefa difícil, pois é necessário estimar tanto

os parâmetros desconhecidos do modelo quanto as variáveis omissas. Isso deu a motivação necessária para a construção do método EM. A derivação das equações abaixo foram adaptadas de Kung *et al.* (2004) para nosso contexto, para mais informações consultar o material original.

Podemos estimar que a fdp condicional para $\mathbf{x}_M$ como:

$$f(\mathbf{x}_M|\mathbf{x}_O,\beta) = \frac{f(\mathbf{x}_M,\mathbf{x}_O|\beta)}{f(\mathbf{x}_O|\beta)}, \tag{2.13}$$

então a log-verossimilhança seria dada por:

$$\begin{aligned} l(\mathbf{x}_O|\beta) \equiv logf(\mathbf{x}_O|\beta) &= logf(\mathbf{x}_i|\beta) - logf(\mathbf{x}_M|\beta,\mathbf{x}_O) \\ &= \mathbb{E}_{\mathbf{x}_M}\left[logf(\mathbf{x}_M,\mathbf{x}_O|\beta)|\mathbf{x}_O,\beta\right] \\ &- \mathbb{E}_{\mathbf{x}_M}\left[logf(\mathbf{x}_M|\mathbf{x}_O,\beta)|\mathbf{x}_O,\beta\right] \end{aligned} \tag{2.14}$$

Para simplificar a notação utiliza-se as função:

$$Q(\beta|\beta^t,\mathbf{x}_O) = \mathbb{E}_{\mathbf{x}_M}\left[logf(\mathbf{x}_M,\mathbf{x}_O|\beta)|\mathbf{x}_O,\beta^t\right]) \tag{2.15}$$

EM é um método para achar a máxima verossimilhança local, que se dá em duas etapas alternadas até a sua convergência. A etapa E é o passo em que se dá a estimação dos atributos faltantes, enquanto a etapa M é o passo de maximização da verossimilhança dos parâmetros do modelo estimado na etapa E.

1. Escolher algum conjunto de parâmetros β^t inicial (t=0)
2. Passo E: Calcular $Q(\beta|\beta^t,\mathbf{x}_O)$
3. Passo M: Encontrar a maximização do log-verossimilhança para os parâmetros estimados $\beta^{t+1} \leftarrow \underset{\beta}{argmax}\ Q(\beta|\beta^t,\mathbf{x}_O)$
4. Repetir passo 2 e 3 até satisfazer os critérios de convergência

a prova da convergência do algoritmo se encontra em Wu (1983).

Neste trabalho aplicamos uma variante do método EM, o Expectation Conditional Maximization (ECM) (MENG; RUBIN, 1993). Assim como o EM o ECM é iterativo, porém os passos executados serão o passo E e o passo CM. No passo E o algoritmo permanece o mesmo, ou seja, estima a função que será maximizada no passo CM. Porém no passo substituído, executamos uma série de maximizações computacionalmente mais simples. Estas maximizações são condicionadas em alguns dos parâmetros (ou em funções dos mesmos) (MCLACHLAN; KRISHNAN, 2008).

2.3 Resumo do capítulo

Neste capítulo foi abordado uma pequena revisão do conteúdo teórico e técnicas necessárias para a formulação do método proposto. Inicialmente introduzimos a regressão linear, explicamos brevemente do que se trata a regularização e como geramos modelos esparsos através do algoritmo RFS. Em seguida foi realizada a exposição do problema de dados faltantes, quais seus mecanismos e formas de tratamento comuns ao problema.

3 MÉTODO PROPOSTO

Neste capítulo é proposto o método de Regressão Forward Stagewise para Dados Faltantes (RFSDF) como uma modificação do RFS, porém robusto ao problema de dados faltantes. Para isso, na seção 3.1 revisitaremos o método RFS apresentado no algoritmo 1 e iremos ponderar sobre as modificações necessárias. Na seção 3.2, apresentaremos o método proposto e as modificações sugeridas. Na seção 3.2.1 é visto como se realiza o cálculo da IMC para as suposições feitas na construção do método.

3.1 Forward Stagewise Regression Revisitado

O algoritmo 1 tem como entrada um par (X,Y), tal que $X \in \mathbb{R}^{n\times p}$ e $Y \in \mathbb{R}^n$, sendo n o número de exemplos e p a quantidade de características no conjunto de treinamento. Neste texto é utilizado a notação $\mathbf{x}_i$ para representar a linha i da matriz X e X_j para representar a coluna j do mesmo modo. Como passo inicial, normalizamos as colunas X_j, para $j = 1,\cdots,p$, e centralizamos Y. Em seguida inicializamos o vetor $\theta = [\theta_1,\cdots,\theta_p]$ com um vetor nulo e o vetor r com Y . Note que estamos tentando obter a seguinte igualdade:

$$Y = X\theta + \mathbf{r}, \tag{3.1}$$

nela, desejamos que o resíduo $\mathbf{r}$ tenha a menor norma euclidiana quadrada possível enquanto mantém o maior número de valores de θ iguais a zero. Portanto o algoritmo computa a cada passo qual o coeficiente tem maior correlação com o resíduo, atualiza o peso de coeficiente e recalcula o resíduo (linhas 4 a 8 do algoritmo 1) até que todos os coeficiente não tenham mais correlação com o resíduo. A maneira de calcular a correlação entre o resíduo e o vetor de características é definida por:

$$\text{Cor}(X_j,\mathbf{r}) = \langle X_j,\mathbf{r}\rangle \tag{3.2}$$

A equação acima é facilmente resolvida caso a matriz esteja completa, porém aqui supomos que X tem algumas instancias com um ou mais atributos não observáveis. Nessas condições é necessário que nosso método possa lidar com os dados faltantes de modo a considerar a incerteza destes dados, além disso precisamos realizar algumas suposições tal como as distribuições de probabilidade dos dados completos e dos dados faltosos e seu mecanismo de falta.

3.2 Forward Stagewise para Dados Faltantes

Nosso método tem entrada similar ao método RFS, um par (X,Y), porém algumas instâncias de X possuem dados faltantes. Portanto iremos separar nossos dados entre observáveis e não observáveis da seguinte forma: Sendo $\mathbf{x}_i$ a i-ésima linha da matriz X. Chamaremos de M_i o vetor de índices de $\mathbf{x}_i$ para quais $\mathbf{x}_i$ tem dados faltosos. E de mesma maneira, em O_i estarão os índices para quais $\mathbf{x}_i$ possui valores observáveis. Logo poderemos particionar nosso vetor $\mathbf{x}_i$ da seguinte forma $\mathbf{x}_i = [\mathbf{x}_{i,M}, \mathbf{x}_{i,O}]$, seguindo a seguinte regra: $\mathbf{x}_{i,j} \in \mathbf{x}_{i,M}$ sse $j \in M$, caso contrário $\mathbf{x}_{i,j} \in \mathbf{x}_{i,O}$. Conseguinte a isto, precisamos supor qual mecanismo coordena a probabilidade de um dado ser oculto. Neste trabalho presumiremos que o padrão de perda dos dados é MAR, pois MCAR é muito restritivo e de acordo com vários autores bastante improvável de acontecer no mundo real (DING; SIMONOFF, 2010). MNAR também foi uma opção descartada pois nele contém uma relação mais complexa normalmente precisando da elaboração de um modelo do próprio mecanismo. Nele, a hipóteses de algum dado ser faltante, deve ser condicionada nos dados observáveis $(P(M_i|\mathbf{x}_i) = P(M_i|\mathbf{x}_{i,O}), \forall i = 1, \cdots, n)$.

A partir da entrada dada, objetivamos obter um vetor de pesos que seja uma representação aceitável do nosso modelo. A etapa que atualiza o peso em direção a convergência, depende diretamente do cálculo da correlação visto na equação 3.2. Portanto calcularemos o valor esperado desta correlação da seguinte maneira:

$$\mathbb{E}\big[\mathrm{Cor}(X_j, \mathbf{r})\big] = \mathbb{E}\big[\langle X_j, \mathbf{r}\rangle\big], \tag{3.3}$$

uma vez que $\mathbf{r} = Y - X\theta$ podemos utilizar a combinação linear de sua esperança para expandir a equação 3.3 de forma que:

$$\begin{aligned}
\mathbb{E}\big[\mathrm{Cor}(X_j, \mathbf{r})\big] &= \mathbb{E}\big[\langle X_j, Y - X\theta\rangle\big] \\
&= \mathbb{E}\big[\langle X_j, Y\rangle\big] - \mathbb{E}\big[\langle X_j, X\theta\rangle\big] \\
&= \sum_{i=1}^{n} (y_i \cdot \mathbb{E}[x_{i,j}]) - \sum_{i=1}^{n} (\mathbb{E}[x_{i,j}\mathbf{x}_i\theta]) \\
&= \sum_{i=1}^{n} (y_i \cdot \mathbb{E}[x_{i,j}] - \mathbb{E}[x_{i,j}\mathbf{x}_i\theta]) \\
&= \sum_{i=1}^{n} \Big(y_i \cdot \mathbb{E}[x_{i,j}] - (\mathbb{E}[x_{i,j}]\mathbb{E}[\mathbf{x}_i\theta] + \mathrm{Cov}[x_{i,j}, \mathbf{x}_i\theta]) \Big) \\
&= \sum_{i=1}^{n} \Big(y_i \cdot \mathbb{E}[x_{i,j}] - \big(\mathbb{E}[x_{i,j}] \sum_{k=1}^{p} \theta_k \mathbb{E}[x_{i,k}] + \sum_{k=1}^{p} \theta_k \mathrm{Cov}[x_{i,j}, x_{i,k}]\big) \Big) \\
&= \sum_{i=1}^{n} \Big(y_i \mathbb{E}[x_{i,j}] - \big(\sum_{k=1}^{p} \theta_k (\mathbb{E}[x_{i,k}]\mathbb{E}[x_{i,j}] + \mathrm{Cov}[x_{i,j}, x_{i,k}])\big) \Big).
\end{aligned} \tag{3.4}$$

Também podemos reescrever o resultado da equação 3.4 de maneira simplificada para observar algumas de suas características,

$$\begin{aligned}\mathbb{E}\big[\mathrm{Cor}(X_j,\mathbf{r})\big] &= \mathbb{E}\big[X_j\big]^T Y - \mathbb{E}\big[X_j\big]^T \mathbb{E}\big[X\big]\theta - \sum_{i=1}^{N} \mathbf{e}_j^T \mathrm{Cov}[\mathbf{x}_i]\theta \\ &= \langle \mathbb{E}\big[X_j\big] Y - \mathbb{E}\big[X\big]\theta \rangle - \sum_{i=1}^{N} \mathbf{e}_j^T \mathrm{Cov}[\mathbf{x}_i]\theta\end{aligned} \tag{3.5}$$

em que $\mathbf{e}_j$ é o j-ésimo vetor da base canônica de $\mathbb{R}^p$ e $\mathrm{Cov}[\mathbf{x}_i]$ é a matriz de covariância relativa a linha i de X. Ao analisar a equação, podemos ver exatamente onde a incerteza é contabilizada. Digamos que ao realizar a imputação das variáveis em X obtemos a matriz $\hat{X} = \mathbb{E}[X]$. Logo a correlação entre alguma característica $\hat{X}_j$ de $\hat{X}$ e seu resíduo é dada por $\mathrm{Cor}(\hat{X}, \hat{\mathbf{r}}) = \langle \hat{X}_j \hat{Y} - \hat{X}\hat{\theta} \rangle$. Logo a incerteza é contabilizada através da covariância, pois se o somatório da covariância é nulo a seguinte equivalência é satisfeita:

$$\mathbb{E}\big[\mathrm{Cor}(X_j,\mathbf{r})\big] = \mathrm{Cor}(\mathbb{E}\big[X_j\big],\mathbf{r}) \tag{3.6}$$

.

A equação 3.4 mostra que a Esperança da correlação é função de $\mathbb{E}[x_{i,j}]$ e $\mathrm{Cov}[x_{i,j}, x_{i,k}]$, para todo $i = 1, \cdots, n$ e para todo $j, k = 1, \cdots, p$. Como não dispomos da distribuição das variáveis aleatórias e precisamos de uma modelagem dos dados, é razoável estima-las a partir dos dados observáveis. Logo, aferimos estas variáveis do seguinte modo:

$$\mathbb{E}[x_{i,j}] = \begin{cases} \mathbb{E}[x_{i,j}|\mathbf{x}_O] & \text{se } x_{i,j} \in \mathbf{x}_M, \\ x_{i,j} & \text{Caso contrário} \end{cases} \tag{3.7}$$

$$\mathrm{Cov}[x_{i,j}, x_{i,k}]) = \begin{cases} \mathrm{Cov}[x_{i,j}, x_{i,k}, |\mathbf{x}_O] & \text{se } x_{i,j}, x_{i,k} \in \mathbf{x}_M, \\ 0 & \text{Caso contrário} \end{cases} \tag{3.8}$$

É importante notar que, caso todos os componentes fossem entradas observáveis, nosso caso se tornaria trivial, pois somente seria preciso substitui-los na expressão final. Porem, supondo que há registros com dados faltantes, temos que utilizar algum método que estime as médias e as variâncias condicionadas nas variáveis observadas. Para conseguir estas estimativas iremos supor que os dados estão na forma de uma gaussiana multivariada e as obteremos pelo método IMC. Tal suposição nos habilita a estimar sua distribuição conjunta, pois "uma importante

propriedade da distribuição gaussiana multivariada é que, se dois conjuntos de variáveis são uma gaussiana conjunta, então a distribuição condicional de um conjunto, condicionado no outro, é também gaussiana" (BISHOP, 2006, p. 85, tradução do autor).

3.2.1 Estimação da esperança e covariância condicional

De maneira semelhante a que consideramos $\mathbf{x}_i = [\mathbf{x}_M, \mathbf{x}_O]$, podemos representar o vetor de média μ_i e sua matriz de covariância Σ_i respectivamente por

$$\mu_i = [\mu_{i,M}, \mu_{i,O}] \text{ e } \Sigma_i = \begin{bmatrix} \Sigma_{i,MM} & \Sigma_{i,MO} \\ \Sigma_{i,OM} & \Sigma_{i,OO} \end{bmatrix} \tag{3.9}$$

em que $\mu_{i,M}$ são as médias dos índices com dados faltantes e $\mu_{i,O}$ as médias de índices com dados observáveis. $\Sigma_{i,MM}$ é covariância dos elementos faltosos por todos os elementos faltosos, $\Sigma_{i,MO} = \Sigma_{i,OM}^{\intercal}$ a covariância dos elementos faltosos pelos observáveis e $\Sigma_{i,OO}$ a dos observáveis pelos observáveis.

Com isso posto, podemos agora resolver as equações de média condicional e a variância condicional pelas expressões 3.10 e 3.11

$$\mu_{M|O} = \mu_M + \Sigma_{MO}\Sigma_{OO}^{-1}(\mathbf{x}_O - \mu_O) \tag{3.10}$$

$$\Sigma_{M|O} = \Sigma_{MM} - \Sigma_{MO}\Sigma_{OO}^{-1}\Sigma_{OM} \tag{3.11}$$

e realizar as imputações necessárias por:

$$\mathbb{E}[\mathbf{x}_{i,M}] = \mu_{M|O} \tag{3.12}$$

$$\text{Cov}[\mathbf{x}_{i,M}, \mathbf{x}_{i,M}] = \Sigma_{M|O}. \tag{3.13}$$

O cálculo das médias e da covariância de nossa base de dados X poderá ser inferido de várias forma. Poderíamos por exemplo empregar as técnicas citadas na seção 2.2.1. Porém procurando manter o viés o mais baixo possível do método, utilizamos o ECM.

3.2.2 Algoritmo do método proposto

Finalmente podemos sumarizar o método proposto no seguinte algoritmo:

Algoritmo 2: Regressão Forward Stagewise para Dados Faltantes

início

Utilizar o ECM para estimar média μ e covariancia Σ

para *cada linha* $\boldsymbol{x}_i$ *de* X **faça**

Ache a média condicional $\mu_{M|O} = \mu_M + \Sigma_{i,MO}\Sigma_{i,OO}^{-1}(x_{i,O} - \mu_{i,O})$

Criar vetor $\mathbf{x}_i'$ substituindo os valores faltantes do vetor $\mathbf{x}_i$ por $\mu_{M|O}$

Substituir a linha $\mathbf{x}_i$ na matriz X por $\mathbf{x}_i'$

Achar a matriz de covariância condicional $\Sigma_{i,M|O} = \Sigma_{i,MM} - \Sigma_{i,MO}\Sigma_{i,OO}^{-1}\Sigma_{i,OM}$

fim

$\mathbf{r} = \mathbf{y}$;

$\theta_1, \theta_2, ..., \theta_p = 0$;

Inicializar ε com um valor pequeno maior que 0

enquanto *Não convergir* **faça**

$maxj = 0$;

$maxValorCor = 0$;

para *cada coluna* X_j *de* X **faça**

calcular $valorCor = E[cor(r, X_j)]$ utilizando a equação 3.4

se $|valorCor| > maxValorCor$ **então**

$maxValorCor = |valorCor|$;

$maxj = j$;

fim

fim

atualize $\theta_{maxj} = \theta_{maxj} + \delta$, onde $\delta = \varepsilon * \text{sinal}(cor[r, x_{maxj}])$

$r = r - \delta \mathbf{x}_{maxj}$

fim

fim

3.3 Resumo do capítulo

Neste capítulo propusemos uma variante ao método RFS para dados incompletos e o denominamos RFSDF. Para isto, derivamos a fórmula para a atualização dos coeficientes de regressão, sendo utilizado o valor esperado da correlação de cada vetor de atributos com o resíduo

do modelo. Na fórmula obtida, encontramos expressões que dependem diretamente das variáveis não-observáveis. Portanto, escolhemos inferir essas variáveis a partir da porção observável do vetor. Para isso, supomos que os dados de entrada possuem distribuição normal multivariada e então utilizamos o ECM para obter os parâmetros da distribuição. Então calculamos a média e a covariância condicional e as substituímos na formula.

4 DISCUSSÃO DOS RESULTADOS

Neste capítulo é discutido os resultados gerados pelo método proposto. Na seção 4.1 discorremos sobre quais as bases de dados, métodos e métricas utilizadas na comparação dos resultados. Enquanto na seção 4.2 analisamos sua performance contra estes outros métodos utilizando as métricas estabelecidas.

4.1 Metodologia

Com o objetivo de avaliar a performance da Regressão Forward Stagewise para Dados Faltantes (RFSDF) escolhemos 5 bases de dados do mundo real em Lichman (2013). Nelas realizamos uma série de experimentos e comparamos seu desempenho com métodos comuns para tratamento de dados faltosos. As bases são descritas e divididas em exemplos de treinamento e teste de acordo com a Tabela 1.

Tabela 1 – Descrição das bases de dados

	# Características	# Exemplos de Treinamento	# Exemplos de Teste
Wine	13	100	78
CPU	9	139	70
Cancer	32	129	65
Automobile Price	15	106	53
Forest Fire	4	344	173

Para cada um dos conjuntos de dados utilizados foi empregado o seguinte procedimento: utilizando a parte do conjunto relativa ao treinamento, replicamos o banco de dados 500 vezes de forma integral e em cada uma das cópias apagamos aleatoriamente algumas de suas entradas. Cada vez que o passo anterior é realizado, utilizamos uma porcentagem fixa de dados faltantes (10%, 20%, 30%, 40% e 50% dessa forma). Para cada base de dados gerada por esse processo, aplicamos os métodos RFSDF, LD e o IMC. Sendo os métodos LD e IMC usados na etapa de pré-processamento para em seguida ser gerado um modelo linear pelo RFS enquanto que o RFSDF gera diretamente o modelo linear. Além disso, utilizamos o algoritmo ECM para estimar os parâmetros da distribuição dos dados em RFSDF e no IMC.

Foram utilizados 2 critérios para comparação entre os modelos gerados por cada método. O primeiro é constituído pelo Erro Quadrático Médio (EQM)(equação 4.1) entre o vetor Y a ser predito e o resultado de cada $\hat{Y}$, sendo $\hat{Y}$ as saídas geradas pelos modelos obtidos aplicando cada método. Em seguida comparamos a Média da Diferença Quadrática entre os Coeficientes (MDQC)(equação 4.2) entre os coeficientes de regressão $\hat{\theta}$ dos métodos e o modelo linear do RFS na mesma base de dados, porém sem os dados faltantes representado por θ. Ao comparar o MDQC entre estes vetores, tentamos medir o quão próximo o modelo gerado pelo método RFSDF se aproxima do modelo que o RFS iria criar em uma situação onde o banco de dados é observável. Para ambos, decidimos realizar a medição durante várias etapas do processo de aprendizagem e para escolher os pontos a serem medidos utilizamos o critério da norma dos coeficientes gerados pelo RFS em um cenário ideal (onde não haveria dados faltantes). Porém no EQM geramos uma medição a mais, onde deixamos o algoritmo convergir independente da norma e novamente testamos o resultado dos modelos

$$EQM = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n} \tag{4.1}$$

$$MDQC = \frac{\sum_{j=1}^{m}(\theta_j - \hat{\theta}_j)^2}{m} \tag{4.2}$$

Em alguns casos não foi possível obter um modelo linear utilizando o método LD. Nestes casos, grande parte dos registros eram removidos devido a elevada taxa de dados faltantes, fazendo com que o número de exemplos se aproximasse ao número de variáveis independentes. Este fato tende a se agravar quanto maior é o número de variáveis independentes, principalmente se essa perda de informação está espalhada em muitas variáveis e não concentrada em poucas (DING; SIMONOFF, 2010).

4.2 Tabelas e discussão dos Resultados

A tabela 2 apresenta a média dos EQMs de todas as execuções entre os modelos gerados por cada método e os vetores alvo. Deixamos os métodos convergirem para obter o modelo final de cada um. Pode-se notar que, a medida que a porcentagem de dados faltantes sobe, o erro dos métodos geralmente sobe. Porém, nosso método tem um desempenho melhor, ficando com a menor média dos EQMs em todos os índices. A diferença entre o RFSDF e os outros métodos é menor quando a quantidade de dados faltantes é pequena e fica mais proeminente

a medida que esta quantidade cresce. Além disso, pode-se observar que o erro do LD sobe de forma mais acentuada dentro dos limites em que este é observável.

Tabela 2 – Média dos MSEs entre a a saída de cada modelo linear e a saída alvo variando o número de dados faltantes entre 10% e 50%.

Wine					
	10%	20%	30%	40%	50%
RFSI	6.3404	6.5319	6.7531	7.3600	8.1164
CMI	6.6190	7.5510	10.7476	22.9362	54.9262
LD	12.0314	23.6968	35.8139	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	2.7838e+05	2.8849e+05	2.9230e+05	3.2206e+05	3.2950e+05
CMI	2.8381e+05	3.0676e+05	3.4010e+05	4.5708e+05	6.2148e+05
LD	3.3261e+05	4.1271e+05	7.8900e+05	1.3920e+06	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	4.2238e+08	4.2386e+08	4.1773e+08	4.2275e+08	4.1695e+08
CMI	4.4615e+08	4.9690e+08	7.7811e+08	1.8076e+09	3.4648e+09
LD	1.5076e+09	1.1875e+09	-	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	8.5119e+04	8.2159e+04	8.0475e+04	7.9577e+04	7.7541e+04
CMI	9.2285e+04	9.4305e+04	9.9716e+04	1.1450e+05	1.6995e+05
LD	1.4298e+05	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	6.4320e+05	6.7355e+05	6.8613e+05	6.9128e+05	7.0489e+05
CMI	6.4374e+05	6.7554e+05	6.9017e+05	6.9693e+05	7.1871e+05
LD	6.4787e+05	7.2750e+05	9.2299e+05	1.2284e+06	1.6003e+06

Já nas tabelas 3 à 6 utilizamos a mesma métrica, porém em diferentes momentos. Utilizando o critério de norma máxima do RFS para cada treinamento em um banco de dados sem dados faltantes, podemos utilizar instantes em que o nosso algoritmo passa por frações fixas dessa norma. Em nosso experimento utilizamos 15%, 30%, 45% e 60% da norma máxima para as comparações. Então comparamos da média dos EQMs entre os resultados dos modelos gerados e o rótulo nos exemplos a serem testados nestas novas condições. Quanto mais próximo da norma máxima, maior é a diferença do resultado ideal em todos os métodos. No entanto, podemos notar que o erro do método proposto tende a ser menos acentuado, característica ainda mais perceptiva em quantidades maiores de dados faltantes. Por esse teste podemos perceber que existe uma constância nos resultados independente do valor da norma utilizado.

Utilizando novamente o critério da norma do teste anterior, nas tabelas 7 à 10 iremos realizar a comparação da média das MDQCs. Como pode ser percebido, a diferença entre o vetor de pesos gerados por cada método e o modelo linear ideal (sem dados faltantes) aumenta com a porcentagem de dados faltantes. Novamente, o RFSDF tem a melhor performance, contando com a menor diferença dentre os 3 métodos.

Tabela 3 – Média dos MSEs entre a saída de cada modelo linear e a saída do modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 15% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0252	0.0479	0.0907	0.1400	0.2515
CMI	0.0260	0.0528	0.1062	0.1690	0.2998
LD	0.4222	2.0001	3.6147	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	4.0759e+03	6.4982e+03	8.7891e+03	1.0178e+04	1.2091e+04
CMI	4.1609e+03	6.6568e+03	9.2065e+03	1.0797e+04	1.3000e+04
LD	1.1142e+04	1.8453e+04	3.0777e+04	5.8361e+04	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	1.3778e+06	2.6714e+06	4.8345e+06	6.6800e+06	1.0402e+07
CMI	1.4139e+06	2.9454e+06	5.7238e+06	8.0085e+06	1.2889e+07
LD	2.7127e+07	9.8578e+07	1.3857e+08	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	6.3593e+02	1.4731e+03	2.2977e+03	3.3158e+03	4.3877e+03
CMI	6.5236e+02	1.5485e+03	2.4654e+03	3.7055e+03	5.2511e+03
LD	6.3960e+04	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	3.8275e+01	7.7152e+01	9.9089e+01	1.3612e+02	1.6613e+02
CMI	3.8974e+01	8.0432e+01	1.0785e+02	1.4705e+02	1.8959e+02
LD	3.8236e+01	1.4870e+02	3.3647e+02	4.7485e+02	6.3490e+02

4.3 Conclusão do capítulo

Neste capítulo realizamos uma série de experimentos para comparar a performance do método proposto. Para isso, utilizamos 2 técnicas para o tratamento dos dados faltantes na etapa de pré-processamento e utilizamos o banco de dados gerado por estas técnicas no método RFS. Em seguida, analisamos os resultados obtidos a partir de 2 métricas: EQM e MDQC. Os

Tabela 4 – Média dos MSEs entre a saída de cada modelo linear e a saída do modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 30% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0584	0.1248	0.2343	0.3375	0.5696
CMI	0.0640	0.1517	0.2944	0.4486	0.7438
LD	0.8319	4.8549	11.4526	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	1.1389e+04	2.0506e+04	2.6642e+04	3.1475e+04	3.7881e+04
CMI	1.1989e+04	2.2267e+04	2.9476e+04	3.5908e+04	4.3587e+04
LD	3.5309e+04	5.8468e+04	9.1464e+04	1.8570e+05	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	3.0645e+06	5.2162e+06	9.0657e+06	1.1327e+07	1.6807e+07
CMI	3.2492e+06	5.8626e+06	1.0466e+07	1.3659e+07	2.1050e+07
LD	6.0881e+07	2.6698e+08	-	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	1.5478e+03	3.5326e+03	5.3410e+03	6.9319e+03	8.7961e+03
CMI	1.6159e+03	3.8796e+03	6.1422e+03	9.1113e+03	1.3233e+04
LD	1.3299e+05	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	1.0899e+02	2.2123e+02	2.8407e+02	3.9011e+02	5.1586e+02
CMI	1.1109e+02	2.4208e+02	3.1861e+02	4.6422e+02	6.0750e+02
LD	1.4906e+02	5.2241e+02	1.1395e+03	1.6694e+03	2.2476e+03

resultados dessas comparações mostram que o método proposto tem melhor desempenho médio que os métodos de tratamento: LD e IMC.

Tabela 5 – Média dos MSEs entre a saída de cada modelo linear e a saída do modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 45% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0674	0.1613	0.2995	0.4587	0.8002
CMI	0.0758	0.2023	0.3965	0.6418	1.1368
LD	1.1442	7.0611	-	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	1.5605e+04	2.9480e+04	4.1208e+04	4.9367e+04	6.1678e+04
CMI	1.6747e+04	3.3727e+04	4.8768e+04	6.0930e+04	7.6581e+04
LD	5.8162e+04	9.6371e+04	1.5874e+05	2.7118e+05	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	4.7866e+06	8.0623e+06	1.4155e+07	1.8501e+07	2.7287e+07
CMI	5.1442e+06	9.1955e+06	1.6538e+07	2.1365e+07	3.2375e+07
LD	1.1254e+08	4.0612e+08	-	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	2.4704e+03	5.4344e+03	8.1659e+03	9.0846e+03	1.0893e+04
CMI	2.6814e+03	6.6682e+03	1.0500e+04	1.5447e+04	2.3298e+04
LD	2.3466e+05	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	1.6507e+02	3.2186e+02	4.1830e+02	5.3546e+02	8.2500e+02
CMI	1.7014e+02	3.6556e+02	5.1438e+02	7.5060e+02	1.0437e+03
LD	2.9703e+02	1.0152e+03	2.1457e+03	3.1654e+03	4.2898e+03

Tabela 6 – Média dos MSEs entre a saída de cada modelo linear e a saída do modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 60% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0864	0.2244	0.4358	0.7148	1.2274
CMI	0.0944	0.2595	0.5272	0.8838	1.5915
LD	1.5086	8.6347	-	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	1.4913e+04	2.9225e+04	4.3872e+04	5.6930e+04	7.3025e+04
CMI	1.6351e+04	3.5117e+04	5.5373e+04	7.3516e+04	9.5680e+04
LD	6.5316e+04	1.1713e+05	2.1825e+05	-	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	6.4919e+06	1.2250e+07	2.0888e+07	2.9159e+07	4.0287e+07
CMI	7.0019e+06	1.4013e+07	2.4845e+07	3.3277e+07	4.9817e+07
LD	1.5031e+08	4.7109e+08	-	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	3.2834e+03	6.7217e+03	9.6568e+03	1.1852e+04	1.4177e+04
CMI	3.7701e+03	9.3836e+03	1.5459e+04	2.2687e+04	3.5356e+04
LD	1.8682e+05	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	2.0063e+02	3.9901e+02	5.3767e+02	6.6533e+02	1.1427e+03
CMI	2.0972e+02	4.8392e+02	6.8925e+02	1.0346e+03	1.5474e+03
LD	4.5561e+02	1.4739e+03	3.3348e+03	4.8844e+03	6.6774e+03

Tabela 7 – Média dos MDCQs entre cada modelo linear e os modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 15% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0015	0.0029	0.0058	0.0089	0.0165
CMI	0.0016	0.0031	0.0068	0.0107	0.0194
LD	0.0209	0.0579	0.0822	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	330.5880	509.1208	735.7456	796.2642	860.8637
CMI	336.5682	522.3986	761.5297	832.7201	909.7454
LD	833.9255	1236.9765	1322.1237	1394.2234	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	2.75195e+05	4.91472e+05	8.64200e+05	1.14612e+06	1.69118e+06
CMI	2.79792e+05	5.35822e+05	1.00553e+06	1.34582e+06	2.03725e+06
LD	3.51822e+06	4.68376e+06	5.39818e+06	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	68.1383	157.1110	218.4614	282.9707	322.4907
CMI	68.8974	158.8175	222.5980	293.8232	356.6808
LD	1042.5727	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	0.4677	0.9759	1.1903	1.5881	1.8192
CMI	0.4765	1.0175	1.2997	1.7196	2.0601
LD	0.4252	3.8364	12.6646	23.3338	36.5733

Tabela 8 – Média dos MDCQs entre cada modelo linear e os modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 30% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0028	0.0060	0.0114	0.0168	0.0297
CMI	0.0031	0.0077	0.0151	0.0232	0.0402
LD	0.0345	0.1261	0.2264	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	949.5511	1642.1833	2362.9053	2609.0817	2905.4348
CMI	994.5057	1767.8236	2581.6698	2890.8464	3208.9734
LD	2858.6407	4416.8752	4713.3837	5112.4878	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	4.99136e+05	7.89447e+05	1.37572e+06	1.64525e+06	2.34146e+06
CMI	5.38478e+05	9.24256e+05	1.65136e+06	2.06429e+06	3.04505e+06
LD	6.07145e+06	1.43461e+07	-	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	315.5959	689.3359	879.4547	939.1615	941.2634
CMI	321.0407	714.4983	984.0727	1214.7930	1433.9887
LD	2738.7149	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	1.3064	2.6853	3.1192	4.2199	5.1891
CMI	1.3349	2.9743	3.6451	5.192	6.3540
LD	1.3788	3.7616	7.3096	10.2142	12.6646

Tabela 9 – Média dos MDCQs entre cada modelo linear e os modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 45% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0033	0.0065	0.0125	0.0194	0.0348
CMI	0.0032	0.0087	0.0179	0.0283	0.0520
LD	0.0425	0.1681	-	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	1277.7359	2385.4040	3638.0817	4105.8572	4838.1707
CMI	1370.3084	2707.6797	4261.6337	4913.7103	5758.5741
LD	4757.6309	7546.5486	8578.5215	9597.8471	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	6.39862e+05	1.08062e+06	1.88349e+06	2.19982e+06	3.00759e+06
CMI	6.97030e+05	1.26571e+06	2.29862e+06	2.74333e+06	3.96610e+06
LD	7.66190e+06	1.52016e+07	-	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	815.4529	1432.2428	1835.1505	1571.4583	1542.1451
CMI	859.2785	1783.3939	2393.9583	2772.8872	3309.5005
LD	5932.4867	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	1.8247	3.6323	4.1453	5.2285	7.4526
CMI	1.8937	4.2270	5.4768	8.0466	10.3822
LD	2.3055	6.5959	13.2102	18.6168	23.3338

Tabela 10 – Média dos MDCQs entre cada modelo linear e os modelo linear obtido pelo RFS no mesmo banco de dados. Neste experimento estabelecemos 60% da norma máxima como ponto de comparação

Wine					
	10%	20%	30%	40%	50%
RFSI	0.0033	0.0084	0.0169	0.0262	0.0426
CMI	0.0037	0.0104	0.0221	0.0350	0.0630
LD	0.0511	0.1926	-	-	-
CPU					
	10%	20%	30%	40%	50%
RFSI	1191.6427	2369.9071	3796.9097	4585.2540	5591.5145
CMI	1313.4053	2841.6674	4745.0750	5810.1135	7133.1241
LD	5174.1832	8978.9877	11144.0691	-	-
Automobile Price					
	10%	20%	30%	40%	50%
RFSI	8.20843e+05	1.55785e+06	2.47574e+06	3.08841e+06	3.95091e+06
CMI	8.92548e+05	1.80668e+06	3.03822e+06	3.66187e+06	5.14811e+06
LD	9.34456e+06	1.83650e+07	-	-	-
Cancer					
	10%	20%	30%	40%	50%
RFSI	1471.5130	2002.8079	2187.5142	2280.6016	1995.2143
CMI	1561.4253	2983.8447	4298.5120	4711.2356	5891.7460
LD	7616.6000	-	-	-	-
Forest-Fire					
	10%	20%	30%	40%	50%
RFSI	2.0451	4.1852	4.8350	5.6707	9.1165
CMI	2.1624	5.2306	6.8008	10.2454	14.4056
LD	3.2613	9.3837	20.2777	28.8248	36.5733

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi proposto o método de Regressão Forward Stagewise para Dados Faltantes RFSDF. Essa variante da RFS supõe que os dados tem distribuição normal e a cada iteração calcula um pequeno incremento para o vetor de pesos. Esse incremento é obtido a partir da esperança da correlação do resíduo com cada vetor de características. Para isso, precisamos calcular a esperança e a correlação condicional e utilizamos o método ECM para achar os parâmetros da distribuição.

A RFSDF foi comparada a métodos populares de tratamento de dados faltantes: o LD e o IMC. Foi observado pelas comparações que, apesar da acurácia de todos os métodos diminuírem com a quantidade de dados faltantes, nosso método é o mais próximo de um modelo ideal (aquele onde todos os dados são observáveis).

Além disso, para a construção do método RFSDF, revisamos alguns conceitos no capítulo 2. Este capítulo foi dividido em 2 seções de maior interesse. No primeiro tópico abordamos a regressão linear, vimos a formulação e o treinamento de dois métodos para gerar modelos lineares e como a regularização pode restringir o valor dos pesos destes modelos. Nesse mesmo tópico foi considerado a existência de modelos lineares esparsos e exibido a construção do algoritmo RFS. Dados faltantes foi o tópico da segunda seção. Nele foi apresentada sua definição, mecanismos e tratamento, além de explicar o método EM para inferir parâmetros de uma distribuição numericamente.

5.1 Trabalhos Futuros

O método proposto supõe que os dados tenha uma distribuição de probabilidade normal multivariada. Essa suposição nem sempre é verdadeira então, o método pode ter seu desempenho prejudicado em algumas dessas situações. Para realizar esta verificação iremos testar o método em mais bancos de dados obtidos do mundo real e avaliar sua consistência. Além disso, desenvolveremos outras variações do método para diferentes distribuições. Em particular, o método utilizando mistura de gaussianas já está em desenvolvimento.

Adicionalmente aos trabalhos acima, iremos realizar a formulação de variações de outros métodos de regressão linear que resultem em modelos esparsos, como o LARS (do inglês, Least Angle Regression) desenvolvido por Efron *et al.* (2004). Estas variações tem como

propósito serem robustas a dados faltantes e incorporar a incerteza do processo de estimação dos dados.

REFERÊNCIAS

ABDELLA, M.; MARWALA, T. The use of genetic algorithms and neural networks to approximate missing data in database. In: **IEEE 3rd International Conference on Computational Cybernetics, 2005. ICCC 2005.** [S.l.: s.n.], 2005. p. 207–212.

ACUNA, E.; RODRIGUEZ, C. The treatment of missing values and its effect in the classifier accuracy. **Classification, Clustering and Data Mining Applications**, p. 639–648, 2004. Disponível em: <http://academic.uprm.edu/~{}eacuna/IFCS04r.p>.

AMBROSIUS, W. **Topics in Biostatistics**. Humana Press, 2007. (Methods in Molecular Biology). ISBN 9781597455305. Disponível em: <https://books.google.ca/books?id=EzFFAQAAIAAJ>.

BELANCHE, L.; KOBAYASHI, V.; ALUJA, T. Handling missing values in kernel methods with application to microbiology data. **Neurocomputing**, v. 141, p. 110–116, Oct 2014.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738.

BOLTON, R. J.; HAND, D. J. **Statistical Fraud Detection: A Review**. 2002.

CLARKE, R.; RESSOM, H. W.; WANG, A.; XUAN, J.; LIU, M. C.; GEHAN, E. A.; WANG, Y. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. **Nat Rev Cancer**, Nature Publishing Group, v. 8, n. 1, p. 37–49, jan. 2008. ISSN 1474-175X. Disponível em: <http://dx.doi.org/10.1038/nrc2294>.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. **JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B**, v. 39, n. 1, p. 1–38, 1977.

DING, Y.; SIMONOFF, J. S. An investigation of missing data methods for classification trees applied to binary response data. **J. Mach. Learn. Res.**, JMLR.org, v. 11, p. 131–170, mar. 2010. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1756006.1756012>.

DONG, Y.; PENG, J. Principled missing data methods for researchers. v. 2, p. 222, 12 2013.

EFRON, B.; HASTIE, T.; JOHNSTONE, I.; TIBSHIRANI, R. Least angle regression. **Annals of Statistics**, v. 32, p. 407–499, 2004.

EIROLA, E.; DOQUIRE, G.; VERLEYSEN, M.; LENDASSE, A. Distance estimation in numerical data sets with missing values. **Inf. Sci.**, Elsevier Science Inc., New York, NY, USA, v. 240, p. 115–128, ago. 2013. ISSN 0020-0255. Disponível em: <http://dx.doi.org/10.1016/j.ins.2013.03.043>.

HASTIE, T.; TIBSHIRANI, R.; WAINWRIGHT, M. **Statistical Learning with Sparsity : The Lasso and Generalizations**. [S.l.]: Chapman and Hall/CRC, 2015.

JUNIOR, A. H. de S.; CORONA, F.; MICHE, Y.; LENDASSE, A.; BARRETO, G. A.; SIMULA, O. Minimal learning machine: A new distance-based method for supervised learning. In: ____. **Advances in Computational Intelligence: 12th International Work-Conference on Artificial Neural Networks, IWANN 2013, Puerto de la Cruz, Tenerife, Spain, June 12-14, 2013, Proceedings, Part I**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 408–416. ISBN 978-3-642-38679-4. Disponível em: <http://dx.doi.org/10.1007/978-3-642-38679-4_40>.

KUNG, S.; MAK, M.; LIN, S. **Biometric Authentication: A Machine Learning Approach**. First. Upper Saddle River, NJ, USA: Prentice Hall Press, 2004. ISBN 0131478249.

LICHMAN, M. **UCI Machine Learning Repository**. 2013. Disponível em: <http://archive.ics.uci.edu/ml>.

LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: Item-to-item collaborative filtering. **IEEE Internet Computing**, v. 7, n. 1, p. 76–80, Jan/Feb 2003. ISSN 1089-7801.

LITTLE, R. J. A.; RUBIN, D. B. **Statistical analysis with missing data (second edition)**. [S.l.]: Chichester: Wiley, 2002.

MCLACHLAN, G.; KRISHNAN, T. **The EM algorithm and extensions**. 2. ed. ed. Hoboken, NJ: Wiley, 2008. (Wiley series in probability and statistics). ISBN 978-0-471-20170-0. Disponível em: <http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+52983362X&sourceid=fbw_bibsonomy>.

MENG, X.-L.; RUBIN, D. B. Maximum likelihood estimation via the ECM algorithm: A general framework. **Biometrika**, Oxford University Press, v. 80, n. 2, p. 267–278, jun. 1993. ISSN 1464-3510. Disponível em: <http://dx.doi.org/10.1093/biomet/80.2.267>.

MESQUITA, D. P. P.; GOMES, J. P. P.; JUNIOR, A. H. S. A minimal learning machine for datasets with missing values. In: **Neural Information Processing - 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I**. [s.n.], 2015. p. 565–572. Disponível em: <http://dx.doi.org/10.1007/978-3-319-26532-2_62>.

MITCHELL, T. M. **Machine Learning**. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. 2 p. ISBN 0070428077, 9780070428072.

PEUGH, J. L.; ENDERS, C. K. Missing data in educational research: A review of reporting practices and suggestions for improvement. **Review of Educational Research**, v. 74, n. 4, p. 525–556, 2004. Disponível em: <http://dx.doi.org/10.3102/00346543074004525>.

RUBIN, D. B. **Multiple Imputation for Nonresponse in Surveys**. [S.l.]: Wiley, 1987. 258 p.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society, Series B**, v. 58, p. 267–288, 1994.

WAYMAN, J. C. **Multiple Imputation For Missing Data: What Is It And How Can I Use It?** 2003.

WEISBERG, S. **Applied linear regression**. New York; Chichester: John Wiley Sons, 1980.

WU, C. F. J. On the convergence properties of the EM algorithm. **Ann Statist**, 1983. Disponível em: <citeseer.nj.nec.com/78906.html>.