

João Victor Barbosa Alves

**Seleção Incremental de Variáveis para
Aprendizado de Máquina utilizando Preditor
Linear e Validação Cruzada**

Belo Horizonte, Minas Gerais - Brasil

2018

João Victor Barbosa Alves

Seleção Incremental de Variáveis para Aprendizado de Máquina utilizando Preditor Linear e Validação Cruzada

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus lacinia, erat vitae pulvinar malesuada, dolor enim laoreet lectus, at tincidunt lorem lorem in lorem. Suspendisse sagittis sem ex.

Universidade Federal de Minas Gerais - UFMG

Escola de Engenharia

Programa de Graduação em Engenharia Elétrica

Orientador: Antônio de Pádua Braga

Belo Horizonte, Minas Gerais - Brasil

2018

João Victor Barbosa Alves

Seleção Incremental de Variáveis para Aprendizado de Máquina utilizando Preditor Linear e Validação Cruzada

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus lacinia, erat vitae pulvinar malesuada, dolor enim laoreet lectus, at tincidunt lorem lorem in lorem. Suspendisse sagittis sem ex.

Trabalho aprovado. 8 de novembro de 2018:

Antônio de Pádua Braga
Orientador

Professor
Convidado 1

Professor
Convidado 2

Belo Horizonte, Minas Gerais - Brasil
2018

São as perguntas que não sabemos responder que mais nos ensinam.

Elas nos ensinam a pensar.

Se você dá uma resposta a um homem, tudo o que ele ganha é um fato qualquer.

Mas, se você lhe der uma pergunta, ele procurará suas próprias respostas.

(...)

Assim, quando ele encontrar as respostas, elas lhe serão preciosas.

Quanto mais difícil a pergunta, com mais empenho procuramos a resposta.

Quanto mais a procuramos, mais aprendemos.

Rothfuss, Patrick - O Temor do Sábio - a Crônica do Matador do Rei - Segundo Dia.

Resumo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Palavras-chave: latex. abntex. editoração de texto.

Sumário

1 Introdução

O aumento da capacidade de armazenamento e processamento de dados tem possibilitado o desenvolvimento de sistemas inteligentes através do aprendizado de máquina. Tal desenvolvimento, por sua vez, permitiu avanços em diversas áreas da computação, microeletrônica e sensoriamento.

Hoje, aplicações tais como pesquisas web, sistemas *anti-spam*, reconhecimento de voz, recomendações de produto e diversas outras são frutos desse progresso. Porém a disponibilidade de quantidades massivas de dados, apresenta não só um horizonte vasto de possíveis aplicações, mas também desafios para seleção e processamento desses dados.

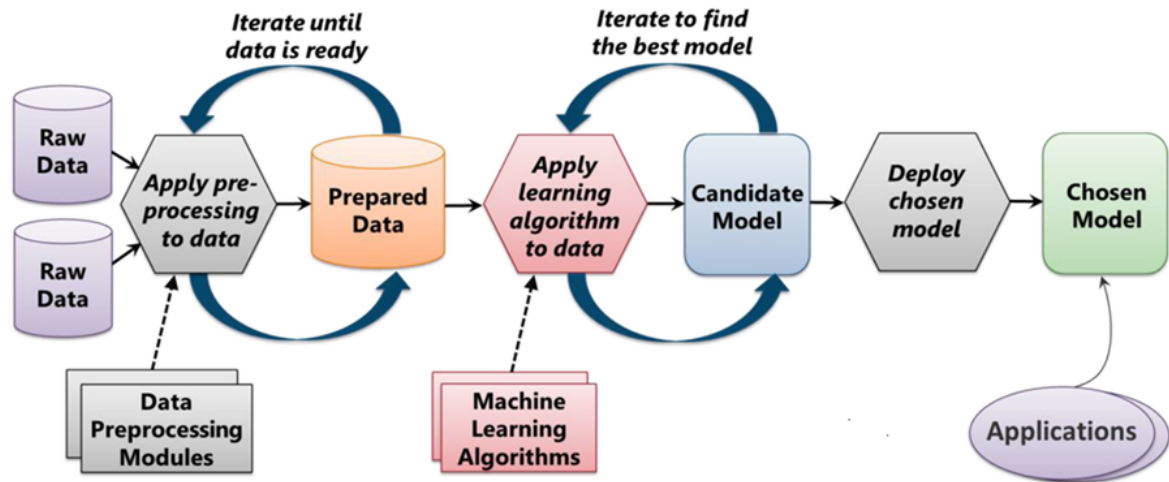
1.1 Aprendizado de Máquina

Aprendizado de máquina é o campo da computação responsável por desenvolver e empregar sistemas ou modelos que, através da sua exposição à experiências, são capazes de melhorar sua performance na realização de determinada tarefa (??).

O processo de desenvolvimento de modelos com aprendizado de máquina pode ser dividido, em linhas gerais, em duas grandes etapas.

Inicialmente os dados disponíveis são analisados e tratados. Nessa etapa procura-se identificar correlações entre as informações disponíveis e as variáveis de interesse. Além disso, são explorados possíveis filtros, transformações e outros algoritmos que possam facilitar o aprendizado do modelo. Também é necessário realizar nessa etapa a separação dos dados que serão utilizados para treinamento, validação e teste no decorrer do processo.

Na etapa seguinte objetiva-se obter um modelo capaz de reproduzir o comportamento do sistema que gerou o conjunto de dados. Para tal, um ou mais modelos e algoritmos são selecionados e o treinamento é realizado. Duas características são de fundamental importância e devem ser controladas: a capacidade de assimilação do comportamento representado pelos dados (complexidade) e a capacidade de extrapolação das saídas para novas entradas (generalização).

Figura 1 – **ADAPTAR**: Fluxo de desenvolvimento de sistemas inteligentes.

Fonte: *Introduction to Microsoft Azure by David Chappell*.

Os algoritmos utilizados nesta segunda etapa podem ser classificados em uma de três categorias, de acordo com as características dos dados disponíveis. (??)

Na categoria de aprendizado não supervisionado, desenvolve-se sistemas capazes identificar padrões implícitos em um conjunto de dados não rotulados.

No aprendizado por reforço, os sistemas se adaptam de acordo com dados de resposta oriundos do ambiente no qual estão envolvidos.

Por fim, no aprendizado supervisionado, estão modelos que, a partir de um conjunto de entradas e saídas conhecidas, são capazes de extrapolar a dinâmica do sistema em questão.

1.2 Generalização

Generalização é o termo usado para descrever a capacidade de um sistema de reagir a novos dados. Isto é, após realizado o treinamento, a capacidade de um sistema de realizar previsões precisas para dados nunca observados.

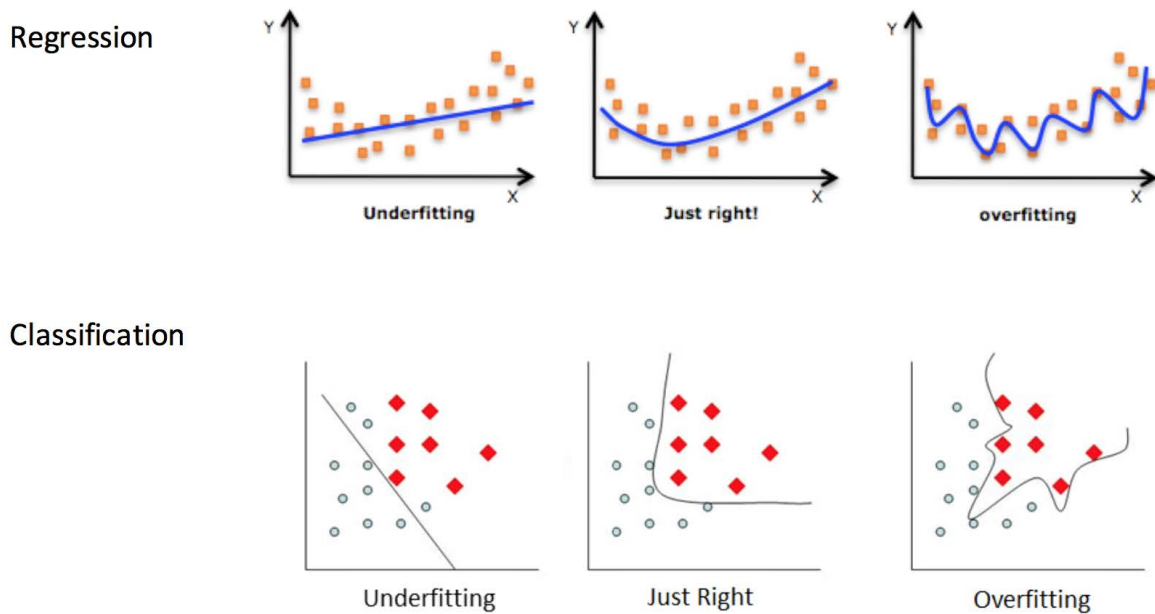
O processo de aprendizado supervisionado é a inferência de um mapeamento de dados de entrada a variáveis de saída a partir de um conjunto de observações. Este pode, portanto, ser interpretado como um ajuste não linear de uma curva (??). Consequentemente, os modelos obtidos através desse processo também estão sujeitos a *overfitting* e *underfitting*.

Overfitting, ou sobre-ajuste, é caracterizado pela alta performance no conjunto de dados de treinamento e baixa performance em dados nunca observados. Isto é, o

modelo assimila desvios causados por erros de medição ou fatores aleatórios presentes no treinamento. Dessa maneira, o erro em relação aos dados de treinamento é reduzido, porém essa melhora não corresponde a uma melhora na representação da realidade.

De maneira similar, o *underfitting*, ou sub-ajuste, é caracterizado pela baixa performance em ambos os conjuntos de dados, indicando que o modelo utilizado não é capaz de representar de maneira satisfatória a complexidade do sistema real.

Figura 2 – Exemplo de underfitting, ajuste ideal e overfitting.



Fonte:

Enquanto a escolha de um modelo adequado é suficiente para evitar o *underfitting*, diversos fatores contribuem e podem contribuir para a ocorrência de *overfitting*.

1.3 Objetivos

Esse trabalho apresenta um método de seleção de variáveis de entrada para modelos treinados através de aprendizado supervisionado.

O algoritmo a ser descrito estima o desempenho das variáveis individualmente e em conjunto através do treinamento de modelos lineares e do cálculo de sua performance aplicando a estratégia de validação cruzada.

2 Fundamentação Teórica

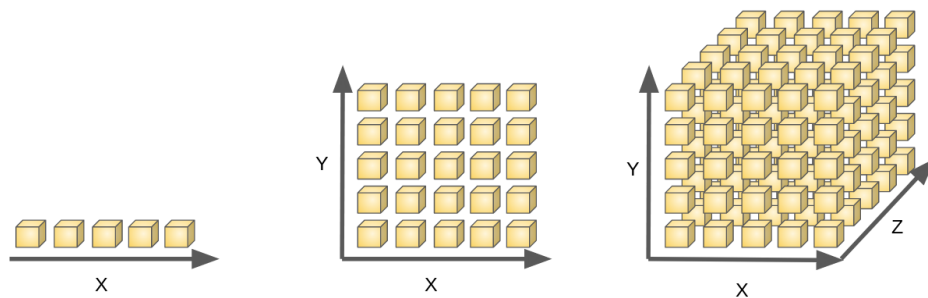
Nesta seção são abordados conceitos fundamentais para a compreensão deste trabalho, assim como tendências atuais da literatura em relação ao tema estudado.

2.1 O problema da dimensionalidade

Sistemas que se utilizam de aprendizado de máquina apresentam respostas baseados nos dados de entrada que lhe são apresentados. Dessa maneira, a determinação de que variáveis são relevantes para o sistema é uma etapa fundamental de seu processo de desenvolvimento.

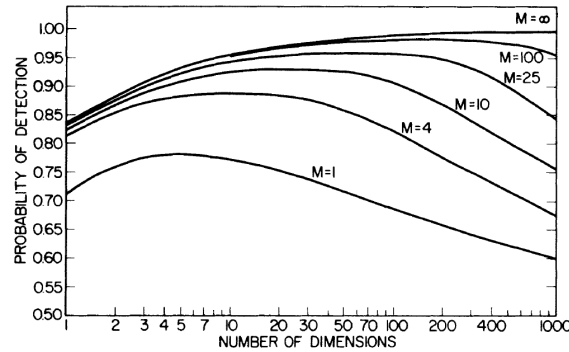
A inclusão de variáveis irrelevantes resulta no aumento da complexidade do modelo, o tornando mais suscetível ao *overfitting* e menos interpretável (??, p. 204). Além disso, um número elevado de variáveis de entrada resulta em um efeito conhecido como a maldição da dimensionalidade (*curse of dimensionality*), onde o aumento do número de dimensões (quantidade de variáveis), implica no aumento exponencial do número de amostras necessárias para representar o espaço de variáveis (??, p. 34).

Figura 3 – Crescimento do espaço de variáveis em virtude do aumento das dimensões.



Fonte: The Curse of Dimensionality (??)

Figura 4 – Performance em relação ao número de dimensões e amostras.



Fonte: A Problem of Dimensionality: A Simple Example (??)

É desejável então que se utilize o menor número possível de variáveis de entrada que permitam ao modelo produzir a resposta correta. Uma maneira de realizar essa redução de dimensionalidade é selecionar um subconjunto das variáveis disponíveis, descartando aquelas que apresentarem pouca ou nenhuma relevância para o problema (??, p. 204).

2.2 Regressão Linear

Os modelos de regressão linear constituem uma classe de modelos que utilizam funções lineares com parâmetros ajustáveis. O exemplo mais simples dessa classe é a função que realiza a combinação linear das entradas com os parâmetros ajustados para gerar uma predição (eq. ??).

$$\hat{y} = w_0 + w_1x_1 + \dots + w_Dx_D = \sum_{i=0}^D w_i\phi(x_i) = \phi(X)W \quad (2.1)$$

Onde D corresponde ao número de dimensões da variável de entrada, $x_0 = 1$ e $\phi(x_i) = x_i$.

Modelos mais complexos e de maior aplicabilidade podem ser obtidos ao se considerar um conjunto fixo de transformações não-lineares ($\phi_n(x)$) ao invés das variáveis originais, ou em conjunto com elas. Esses modelos são lineares em relação às suas variáveis independentes, porém são não-lineares em relação às variáveis de entrada (??).

2.2.1 Treinamento de modelos lineares

O treinamento de modelos lineares consiste em encontrar o vetor de parâmetros W que maximize a similaridade entre o modelo e sistema modelado. Esse treinamento

é normalmente realizado minimizando-se o somatório dos erros quadráticos (eq. ??) em função do vetor W .

$$\begin{aligned}
 E_{sq}(W) &= \frac{1}{2} \sum_{n=1}^N (y_n - \hat{y}_n)^2 \\
 &= \frac{1}{2} (Y - \hat{Y})^T (Y - \hat{Y}) \\
 &= \frac{1}{2} (Y^T Y - 2\hat{Y}^T Y + \hat{Y}^T \hat{Y}) \\
 &= \frac{1}{2} \left(Y^T Y - 2W^T \phi(X)^T Y + \sum_{n=1}^N (\phi(X_n) W)^2 \right)
 \end{aligned} \tag{2.2}$$

$$\begin{aligned}
 \frac{\partial E_{sq}(W)}{\partial W} &= \frac{1}{2} \left(0 - 2\phi(X)^T Y + 2 \sum_{n=1}^N (\phi(X_n)^T \phi(X_n)) W \right) \\
 &= -\phi(X)^T Y + \phi(X)^T \phi(X) W
 \end{aligned} \tag{2.3}$$

Igualando-se a eq. ?? a zero e isolando o vetor W , obtém-se a eq. ??, conhecida como a equação normal para o problema dos mínimos quadrados.

$$\begin{aligned}
 \phi(X)^T \phi(X) W &= \phi(X)^T Y \\
 W &= (\phi(X)^T \phi(X))^{-1} \phi(X)^T Y \\
 W &= \phi(X)^+ Y
 \end{aligned} \tag{2.4}$$

O termo $(\phi(X)^T \phi(X))$ pode se aproximar de uma matriz singular se muitas das variáveis envolvidas forem linearmente dependentes, resultando assim em dificuldades para o cálculo numérico dos valores e possivelmente em um vetor de parâmetros de alta magnitude. Um termo de regularização pode ser adicionado na eq. ?? para minimizar esse problema, garantindo que a nova matriz não é singular, essa técnica ficou conhecida como *Ridge regression*. A dedução da equação normal com termo de regularização é análoga à apresentada e o resulta na eq. ??.

$$W = (\lambda I + \phi(X)^T \phi(X))^{-1} \phi(X)^T Y = A^{-1} \phi(X)_{[i]}^T Y \tag{2.5}$$

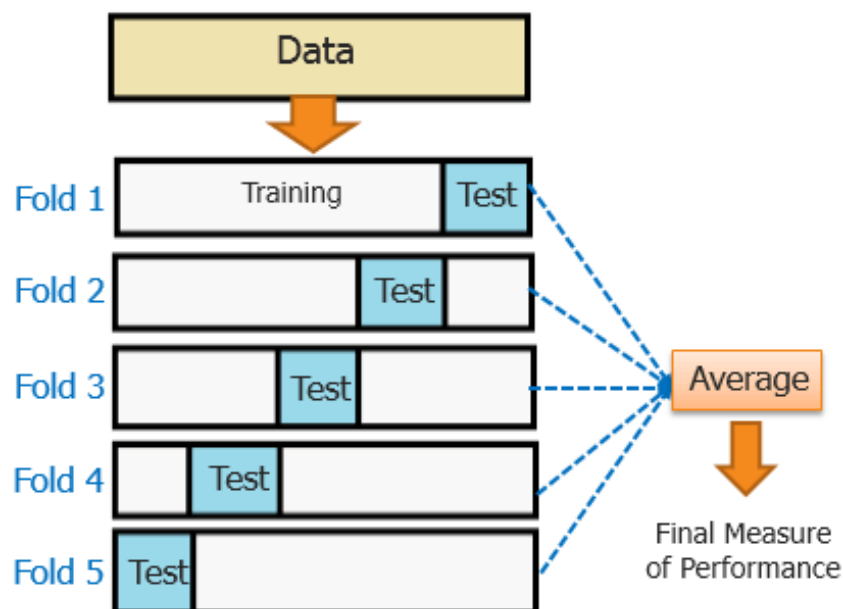
A adição do termo de regularização tem ainda o efeito de limitar a complexidade efetiva do modelo, reduzindo o *overfitting* e possibilitando a utilização de conjuntos de dados menores (??).

2.3 Validação Cruzada

A validação cruzada (*cross-validation*) é um conjunto de metodologias de treinamento e validação que, de maneira simples e efetiva, permitem a estimativa do erro de generalização do modelo.

No método *k-fold* de validação cruzada, os dados de treinamento são divididos em k conjuntos aleatórios de tamanhos aproximadamente iguais. Realizada essa divisão, a cada iteração do treinamento, os parâmetros do modelo são determinados utilizando $k-1$ conjuntos e sua performance é avaliada no conjunto restante. O processo se repete até que todos os k conjuntos tenham sido utilizados para avaliação e a média das performances observadas fornecem uma estimativa da performance de generalização do modelo (??).

Figura 5 – Validação cruzada: *5-fold*.



Fonte: *Different types of Validations in Machine Learning (Cross Validation)* (??)

Um caso especial do método descrito é o *leave-one-out cross-validation* (LOOCV), onde k é igual ao total de amostras disponíveis (N), ou seja, o treinamento é realizado N vezes com $N-1$ amostras, e validado na amostra excluída. Tal abordagem apesar de apresentar, em geral, um custo computacional elevado, resulta em um modelo com menor variabilidade e viés para modelos de regressão linear (??), produzindo assim resultados com melhor performance e menos *overfitting*.

2.3.1 *Leave-one-out Cross-validation* e Regressão Linear

Especificamente para o caso do preditor linear, é possível determinar o resultado do LOOCV sem a necessidade de se calcular N preditores, tornando essa uma estratégia interessante para determinar a capacidade de generalização de um modelo linear. A seguir uma dedução adaptada de (??, p. 268) é apresentada.

O erro de LOOCV corresponde a média dos erros quadráticos de cada estimador obtido excluindo-se uma amostra.

$$LOOCV = \frac{1}{N} \sum_{i=1}^N e_{[i]}^2 \quad (2.6)$$

$$e_{[i]} = y_i - \hat{y}_{[i]} = y_i - x_i^T W_{[i]} \quad (2.7)$$

Onde o i indexa a amostra excluída no treinamento, $\hat{y}_{[i]}$ corresponde à saída do preditor calculado sem a amostra (x_i, y_i) e $W_{[i]}$ os parâmetros desse preditor.

Pode-se definir o vetor de parâmetros $W_{[i]}$ conforme a eq. ??.

$$W_{[i]} = A_{[i]}^{-1} \phi(X)_{[i]}^T Y_{[i]} \quad (2.8)$$

É possível perceber as seguintes igualdades:

$$\begin{aligned} A_{[i]} &= \lambda I + \phi(X)_{[i]}^T \phi(X)_{[i]} \\ &= \lambda I + \phi(X)^T \phi(X) - x_i^T x_i \\ &= A - x_i^T x_i \end{aligned} \quad (2.9)$$

$$\phi(X)_{[i]}^T Y_{[i]} = \phi(X)^T Y - x_i^T y_i \quad (2.10)$$

Aplicando a formula de Sherman–Morrison à eq. ??, temos:

$$\begin{aligned} h_i &= x_i A^{-1} x_i^T \\ A_{[i]}^{-1} &= A^{-1} + \frac{A^{-1} x_i x_i^T A^{-1}}{1 - h_i} \end{aligned} \quad (2.11)$$

Substituindo as eq. ?? e ?? na eq. ??.

$$\begin{aligned} W_{[i]} &= \left(A^{-1} + \frac{A^{-1} x_i^T x_i A^{-1}}{1 - h_i} \right) (\phi(X)^T Y - x_i^T y_i) \\ &= W - A^{-1} x_i^T y_i + \frac{A^{-1} x_i^T x_i W}{1 - h_i} - \frac{A^{-1} x_i^T x_i A^{-1} x_i^T y_i}{1 - h_i} \\ &= W - \frac{A^{-1} x_i^T}{1 - h_i} (y_i (1 - h_i) - x_i^T W + h_i y_i) \\ &= W - \frac{A^{-1} x_i^T}{1 - h_i} (y_i - \hat{y}_i) \end{aligned} \quad (2.12)$$

Substituindo a eq. ?? na eq. ??.

$$\begin{aligned}
 e_{[i]} &= y_i - x_i W_{[i]} \\
 &= y_i - x_i \left(W - \frac{A^{-1} x_i^T}{1 - h_i} (y_i - \hat{y}_i) \right) \\
 &= y_i - \hat{y}_i + \frac{h_i}{1 - h_i} (y_i - \hat{y}_i) \\
 &= \frac{y_i - \hat{y}_i}{1 - h_i}
 \end{aligned} \tag{2.13}$$

Seja P a matriz de projeção (??, p. 303) do preditor (também conhecida como matriz chapéu (??, p. 266)) e a matriz de aniquilação M (??, p. 18) definidas conforme a eq. ??

$$\begin{aligned}
 \hat{Y} &= PY \implies P = XA^{-1}X^T \\
 MY &= Y - \hat{Y} \implies M = I - P = I - XA^{-1}X^T
 \end{aligned} \tag{2.14}$$

Para cada índice i , o numerador da eq. ?? corresponde a linha de mesmo índice da matriz MY . Além disso o denominador da equação corresponde ao elemento da diagonal da matriz M de mesmo índice. A partir dessas observações, pode-se agrupar todos os valores de $e_{[i]}$ na forma da matriz E_{LOOCV} , conforme a equação ??.

$$E_{LOOCV} = \text{diag}(M)^{-1}MY \tag{2.15}$$

Como a matriz M é simétrica, a eq. ?? pode ser reescrita conforme eq. ??.

$$\begin{aligned}
 LOOCV &= \frac{1}{N} \sum_{i=1}^N e_{[i]}^2 \\
 &= \frac{1}{N} E_{LOOCV}^2 \\
 &= \frac{1}{N} E_{LOOCV}^T E_{LOOCV} \\
 &= \frac{1}{N} Y^T M \text{diag}(R)^{-2} MY
 \end{aligned} \tag{2.16}$$

2.4 Aprendizado Incremental

A atualização dos parâmetros do modelo de regressão linear para uma nova amostra pode ser realizada de maneira incremental, sem a necessidade de se calcular um novo modelo (??). Nesse trabalho porém, é de interesse calcular a atualização incremental ao se adicionar uma nova variável. Em especial, é de interesse a determinação de uma forma fechada para a matriz de aniquilação, usada no calculo do erro de validação cruzada.

A matrizes de interesse para o problema de dimensão $p + 1$ tem a forma descrita na eq. ??

$$\begin{aligned} X_{p+1} &= [\mathbf{X}_{\mathbf{p}} \ x_{p+1}] \\ Y_{p+1} &= [\mathbf{Y}_{\mathbf{p}} \ y_{p+1}] \\ A_{p+1} &= (X_{p+1}^T X_{p+1} + \lambda I) = \begin{bmatrix} \mathbf{A}_{\mathbf{p}} & \mathbf{X}_{\mathbf{p}}^T x_{p+1} \\ x_{p+1}^T \mathbf{X}_{\mathbf{p}} & x_{p+1}^T x_{p+1} + \lambda \end{bmatrix} \\ M_{p+1} &= I - X_{p+1} A_{p+1}^{-1} X_{p+1}^T \end{aligned} \quad (2.17)$$

É necessário então determinar a matriz A_{p+1}^{-1} em função das matrizes já conhecidas. Para tal, aplica-se a identidade da inversa de uma matriz em blocos, conforme a eq. ??, na matriz A_{p+1} .

$$\begin{aligned} \Delta &= (\mathbf{D} - \mathbf{C} \mathbf{A}^{-1} \mathbf{B}) \\ \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{A}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{A}^{-1} \mathbf{B} \Delta^{-1} \mathbf{C} \mathbf{A}^{-1} & -\mathbf{A}^{-1} \mathbf{B} \Delta^{-1} \\ -\Delta^{-1} \mathbf{C} \mathbf{A}^{-1} & \Delta^{-1} \end{bmatrix} \end{aligned} \quad (2.18)$$

Através da eq. ??, obtém-se os termos A , B , C , D e Δ na eq. ??.

$$\begin{aligned} A &= A_p \\ B &= X_p^T x_{p+1} \\ C &= x_{p+1}^T X_p = B^T \\ D &= \lambda + x_{p+1}^T x_{p+1} \\ \Delta &= \lambda + x_{p+1}^T x_{p+1} - x_{p+1}^T X_p A_p^{-1} X_p^T x_{p+1} \end{aligned} \quad (2.19)$$

O termo Δ pode ser simplificado para a forma da eq. ??.

$$\begin{aligned} \Delta &= \lambda + x_{p+1}^T x_{p+1} - x_{p+1}^T X_p A_p^{-1} X_p^T x_{p+1} \\ &= \lambda + x_{p+1}^T (I x_{p+1} - X_p A_p^{-1} X_p^T x_{p+1}) \\ &= \lambda + x_{p+1}^T (I - X_p A_p^{-1} X_p^T) x_{p+1} \\ &= \lambda + x_{p+1}^T M_p x_{p+1} \end{aligned} \quad (2.20)$$

Substituindo, obtém-se a eq. ??

$$\begin{aligned} A_{p+1}^{-1} &= \begin{bmatrix} \mathbf{A}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\Delta} \begin{bmatrix} \mathbf{A}^{-1} \mathbf{B} \mathbf{C} \mathbf{A}^{-1} & -\mathbf{A}^{-1} \mathbf{B} \\ -\mathbf{C} \mathbf{A}^{-1} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_{\mathbf{p}}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\Delta} \begin{bmatrix} A_p^{-1} (X_p^T x_{p+1}) (X_p^T x_{p+1})^T A_p^{-1} & -A_p^{-1} (X_p^T x_{p+1}) \\ -(X_p^T x_{p+1})^T A_p^{-1} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_{\mathbf{p}}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\Delta} \begin{bmatrix} A_p^{-1} (X_p^T x_{p+1}) \\ -1 \end{bmatrix} \begin{bmatrix} x_{p+1}^T X_p A_p^{-1} & -1 \end{bmatrix} \end{aligned} \quad (2.21)$$

Por fim, substituindo a eq. ?? no termo M_{p+1} da eq. ??, obtém-se a expressão para a nova matriz de aniquilação.

$$\begin{aligned}
M_{p+1} &= I - X_{p+1} A_{p+1}^{-1} X_{p+1}^T \\
&= I - X_{p+1} \left(\begin{bmatrix} \mathbf{A}_p^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\Delta} \begin{bmatrix} A_p^{-1}(X_p^T x_{p+1}) \\ -1 \end{bmatrix} \begin{bmatrix} x_{p+1}^T X_p A_p^{-1} & -1 \end{bmatrix} \right) X_{p+1}^T \\
&= \left(I - \begin{bmatrix} \mathbf{X}_p & x_{p+1} \end{bmatrix} \begin{bmatrix} \mathbf{A}_p^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}_p^T \\ x_{p+1}^T \end{bmatrix} \right) - \\
&\quad \frac{1}{\Delta} \left(\begin{bmatrix} \mathbf{X}_p & x_{p+1} \end{bmatrix} \begin{bmatrix} A_p^{-1}(X_p^T x_{p+1}) \\ -1 \end{bmatrix} \begin{bmatrix} x_{p+1}^T X_p A_p^{-1} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_p^T \\ x_{p+1}^T \end{bmatrix} \right) \quad (2.22) \\
&= M_p - \frac{1}{\Delta} \left(\mathbf{X}_p A_p^{-1} X_p^T x_{p+1} - x_{p+1} \right) \left(x_{p+1}^T X_p A_p^{-1} \mathbf{X}_p^T - x_{p+1}^T \right) \\
&= M_p - \frac{1}{\Delta} \left(\mathbf{X}_p A_p^{-1} X_p^T - I \right) x_{p+1} x_{p+1}^T \left(X_p A_p^{-1} \mathbf{X}_p^T - I \right) \\
&= M_p - \frac{M_p x_{p+1} x_{p+1}^T M_p}{\Delta} \\
M_{p+1} &= M_p - \frac{M_p x_{p+1} x_{p+1}^T M_p}{\lambda + x_{p+1}^T M_p x_{p+1}}
\end{aligned}$$

3 Desenvolvimento e Metodologia

3.1 Algoritmo

3.1.1 Seleção de Variáveis *Stepwise*

O algoritmo proposto nesse trabalho é baseado no método de seleção de variáveis *stepwise* (*Stepwise Selection*), comumente utilizado em conjunto com modelos de regressão linear. É um método de *greedy search*, onde, a cada iteração, a variável que apresentar o melhor ganho de performance é adicionada ao conjunto de entradas.

O modelo é construído incrementalmente até que não haja mais melhora de performance ao acrescentar alguma das variáveis restantes ou não haja mais variáveis para serem consideradas. O método é descrito em pseudocódigo no algoritmo ??

Algoritmo 1: *Forward Stepwise Selection*

Entrada: *variáveis*: lista contendo as variáveis disponíveis;

Entrada: *saídas*: lista contendo as saídas do modelo;

Saída: *selecionadas*: lista de variáveis relevantes.

selecionadas $\leftarrow \{ \}$

melhorErro $\leftarrow \infty$

repita

melhorVariável $\leftarrow \text{NULL}$

para cada *elemento var* em *variáveis* **faça**

entradas $\leftarrow (var \cup selecionadas)$

model $\leftarrow \text{ajuste}(entradas, saídas)$

erro $\leftarrow \text{avalia}(\text{model}, entradas, saídas)$

se *erro* < *melhorErro* **então**

melhorErro $\leftarrow erro$

melhorVariável $\leftarrow var$

selecionadas $\leftarrow (melhorVariável \cup selecionadas)$

variáveis $\leftarrow (variáveis \setminus \{melhorVariável\})$

até *variáveis* == $\{ \}$ ou (critério de parada);

Esse algoritmo apresenta uma redução expressiva na quantidade de modelos ajustados em relação ao método da seleção do melhor subconjunto ($(1 + (p^2 + p)/2)$ vs. 2^p), porém não há garantia que o subconjunto determinado é a combinação ótima das variáveis (??, p. 208).

A utilização de uma métrica de avaliação que considere simplesmente os dados ajustados não é adequada, uma vez que, nessa situação, o incremento de uma variá-

vel no modelo sempre acarretará em uma melhora no erro de treinamento, porém não necessariamente na capacidade de generalização do modelo.

Algumas técnicas tentam determinar o erro de generalização através do ajuste do erro de treinamento, tais como o critério de Informação de Akaike (AIC) ou critério de informação Bayesiano (BIC), porém elas se baseiam no comportamento assintótico, isto é, quando a quantidade de amostras é bastante elevada.

Uma alternativa a essas técnicas é a utilização de validação cruzada, onde o modelo selecionado é aquele que apresenta a melhor performance no conjunto de testes. Dessa maneira, obtém-se diretamente uma estimativa do erro de generalização, além de se assumir menos condições em relação ao modelo e aos dados utilizados. O grande desafio dessa técnica é o custo computacional da validação cruzada que, quando associado ao custo do método de seleção de variáveis *stepwise* pode tornar proibitiva sua implementação.

Em especial, a utilização desse método associada ao *leave-one-out cross-validation* resultaria no ajuste de $1/2(p^2 + p + 2)(N - 1)$ modelos. Tornando o custo do algoritmo inviável mesmo com o crescimento .

3.1.2 Método Incremental

Para minimizar o problema do custo computacional, o método implementado utiliza o resultado apresentado na eq. ?? para calcular o erro de validação cruzada sem a necessidade de calcular $N - 1$ modelos de regressão linear. Além disso, o cálculo da matriz de aniquilação é realizado de maneira incremental, conforme deduzido na eq. ??.

O algoritmo ?? é então modificado para realizar o cálculo incremental, utilizando

LOOCV como métrica para seleção dos das variáveis e atualização incremental do modelo.

Algoritmo 2: *Forward Stepwise Incremental Selection*

Entrada: *variáveis*: lista contendo as variáveis disponíveis;

Entrada: *saídas*: lista contendo as saídas do modelo;

Saída: *selecionadas*: lista de variáveis relevantes.

$selecionadas \leftarrow \{ \}$

$melhorErro \leftarrow \infty$

$M \leftarrow I_N$

repita

$melhorVariável \leftarrow NULL$

para cada *elemento var em variáveis* **faça**

$M' \leftarrow ajusteIncremental(M, var)$

$erro \leftarrow erroLOOCV(M, saídas)$

se $erro < melhorErro$ **então**

$melhorErro \leftarrow erro$

$melhorVariável \leftarrow var$

$melhorM \leftarrow M'$

$M \leftarrow melhorM$

$selecionadas \leftarrow (melhorVariável \cup selecionadas)$

$variáveis \leftarrow (variáveis \setminus \{melhor_{variável}\})$

até $variáveis == \{ \}$ *ou (critério de parada)*;

3.2 Testes e Validação

3.2.1 Conjuntos de dados

Para teste e validação do algoritmo desenvolvido, utilizou-se três bancos de dados conhecidos na literatura, especificados na tabela ??.

Tabela 1 – Conjuntos de dados utilizados para validação.

Nome	Variáveis	Amostras
<i>Forest Fires</i>	12	517
<i>USA Housing Dataset</i>	80	1460
<i>Wiscoin Breast Cancer</i>	32	194

3.2.2 Metodologia de Validação

Para verificar o funcionamento do algoritmo, variáveis falsas foram geradas e anexadas aos três conjuntos de dados indicados na ??. As variáveis falsas utilizadas foram amostradas através de distribuições aleatórias, descritas na.(TODO: Criar tabela)

4 Resultados e Discussão

5 Conclusão