Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Classificação Binária pelo Método dos Mínimos Quadrados

Andrei Albani[1]
ITA, São José dos Campos, SP
Lucca Moura Zoppi[2]
ITA, São José dos Campos, SP

Resumo Neste trabalho, realizou-se a classificação binária de um conjunto de dados envolvendo passageiros do navio naufragado em 1912 *RMS Titanic* com acurácia de 78%. O conjunto de classes foi $\mathscr{C} = \{1, 0\} \equiv \{\text{sobreviveu, não sobreviveu}\}$ e a classificação foi feita a partir da regressão linear com o Método dos Mínimos Quadrados (MMQ) sobre uma matriz de observações modificada por *feature enginnering*, enquanto o mapeamento dos valores estimados pela regressão linear para o conjunto de classes $\mathscr{C}$ foi feito por uma função de transformação com hiperparâmetro otimizado por uma busca exaustiva. O resultado obtido foi similar ao que resulta da aplicação da regressão linear com a biblioteca `LsqFit.jl` em termos de acurácia da classificação sobre o conjunto de testes.

Palavras-chave. Classificação Binária, Regressão Linear, Método dos Mínimos Quadrados, Decomposição em Valores Singulares, *Feature Engineering*

1 Introdução

Uma das tarefas mais importantes e mais requisitadas nos problemas do mundo real é a classificação, isto é, dado uma série de atributos de um objeto, identificá-lo como um dos elementos de um conjunto de classes. O problema de classificação binária consiste em determinar o modelo que, dado um conjunto de indivíduos, possui a maior taxa de acerto na identificação de cada indivíduo com uma das duas possíveis classes de categorização. A determinação de tal modelo é exigida em incontáveis aplicações práticas, a saber: detecção de *spam* em *emails*; diagnóstico de doenças; identificação de fraudes em transações; decodificação de *bits* em sistemas de comunicação e outras [2].

Dentre os métodos existentes para a resolução desse problema, certamente se destacam por sua grande eficácia e robustez o uso de redes neurais [1,5] e o método de regressão logística [4]. Apesar de sua simplicidade, ao ser comparado com os outros métodos citados, o método dos quadrados mínimos também pode ser utilizado na resolução de um problema de classificação binária. Tal método se resume em determinar o estimador que minimiza a soma dos quadrados dos resíduos obtidos em uma regressão linear, isto é, minimiza a soma dos quadrados dos erros dos pontos estimados em relação aos pontos das observações em dado conjunto chamado conjunto de treino.

Neste trabalho, objetiva-se fazer a classificação de indivíduos, listados como passageiros do RMS Titanic, entre "sobrevivente" e "não-sobrevivente" ao naufrágio do navio em 1912, através da aplicação de regressão linear com Método dos Mínimos Quadrados (MMQ) e, então, avaliar o método segundo sua taxa de acerto (acurácia) sobre um conjunto de testes e em comparação a uma implementação de ajuste linear disponível na biblioteca `LsqFit.jl`.

[1]andrei.albani@hotmail.com
[2]lucca.maia@ga.ita.br

2 Desenvolvimento Teórico

Seja o problema de classificação binária com conjunto de classes $\mathscr{C} = \{\alpha_1, \alpha_2\}$ e espaço de observações $\mathbb{R}^n$.

Considera-se $x \in \mathbb{R}^n$ um vetor de entrada e $y \in \mathbb{R}$ o valor de saída, assumindo-se válida uma relação do tipo $y \approx f(x)$, com $f : \mathbb{R}^n \longrightarrow \mathscr{C}$, toma-se o estimador $\hat{f} : \mathbb{R}^n \longrightarrow \mathscr{C}$ tal que $\hat{f}(x) = \hat{y}$.

O estimador pode ser obtido com base em uma função da forma (1), em que $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ e $\theta_i \in \mathbb{R}$. Dessa forma, $\tilde{f} : \mathbb{R}^n \longrightarrow \mathbb{R}$ e é necessária uma transformação $\mathscr{T} : \mathbb{R} \longrightarrow \mathscr{C}$ que mapeie a imagem de $\tilde{f}$ em $\mathscr{C}$, tal que (2).

$$\tilde{f}(x) = \sum_{i=1}^{p} \theta_i f_i(x) \tag{1}$$

$$\hat{f}(x) = \mathscr{T}\{\tilde{f}(x)\} \tag{2}$$

Definindo-se o i-ésimo resíduo como $r^{(i)} = y^{(i)} - \tilde{f}(x^{(i)})$, para um conjunto de dados $x^{(1)}, x^{(2)}, ..., x^{(m)}$ e $y^{(1)}, y^{(2)}, ..., y^{(m)}$, o classificador pelo método dos mínimos quadrados é aquele que minimiza a soma dos quadrados dos resíduos (3).

$$R = \sum_{i=1}^{m} (r^{(i)})^2 \tag{3}$$

Vetorialmente, os dados desse conjunto podem ser expressos por (4).

$$\begin{cases} y^d & = (y^{(1)}, ..., y^{(m)}) \in \mathbb{R}^m \\ r^d & = (r^{(1)}, ..., r^{(m)}) \in \mathbb{R}^m \end{cases} \tag{4}$$

Tomando-se uma matriz $A \in \mathbb{R}^{m \times p}$ cujos elementos obedecem à relação $A_{ij} = \tilde{f}_j(x^{(i)})$, tem-se (5).

$$A\theta = \tilde{y}^d \tag{5}$$

Dessa forma, a soma dos quadrados dos resíduos é dada pela equação (6).

$$\|r^d\|_2^2 = \|y^d - \tilde{y}^d\|_2^2 = \|y^d - A\theta\|_2^2 \tag{6}$$

Em que $\theta = (\theta_1, ..., \theta_p)$, $\theta \in \mathbb{R}^p$.

Portanto, o vetor $\hat{\theta}$ de parâmetros que é a solução do problema de quadrados mínimos [2] é dado por (7).

$$\hat{\theta} = A^\dagger y^d \tag{7}$$

Em que $A^\dagger$ é a pseudo-inversa da matriz A [3].

Considerando-se a decomposição em valores singulares *SVD - Single Value Decomposition* - de A em (8),

$$A = U\Sigma V^T \tag{8}$$

em que $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{p \times p}$ são matrizes ortogonais e $\Sigma \in \mathbb{R}^{m \times p}$ é a matriz retangular diagonal contendo os valores singulares de A [3], a pseudo-inversa $A^\dagger$ pode ser obtida por (9), a partir da decomposição em valores singulares reduzida de A [3].

$$A^\dagger = V_1 \Sigma_1^{-1} U_1^T \tag{9}$$

Em que, dado $posto(A) = r$, $V_1 \in \mathbb{R}^{p\times r}$ é obtida das r primeiras colunas de V, $U_1 \in \mathbb{R}^{m\times r}$ é formada pelas r primeiras colunas de U e $\Sigma_1 \in \mathbb{R}^{r\times r}$ é a matriz diagonal cujos elementos são os valores singulares de A [3].

Por fim, um método simples para se obter $\mathscr{T} : \mathbb{R} \longrightarrow \mathscr{C}$ é escolher um valor fixo $\alpha \in \mathbb{R}$ e definir (10).

$$\mathscr{T}(x) = \begin{cases} \alpha_1, \ x < \alpha \\ \alpha_2, \ x \geq \alpha \end{cases} \tag{10}$$

O valor de α pode ser determinado por uma heurística ou por uma busca sobre um intervalo que resulta em um valor de α tal que se maximiza alguma métrica de performance do modelo [2].

3 Resultados e Discussão

Segundo o que foi descrito em §2, foram implementados em linguagem `Julia` versão `1.9.2` o modelamento, ajuste e predição para o problema de classificação binária descrito em Titanic - Machine Learning from Disaster. A implementação completa com documentação na forma de comentários para cada parte do código se encontra no repositório público do *Github*, disponível em Projeto do Exame de MAT-55.

Além disso, como há dados não numéricos ou faltantes nos conjuntos de observações, é necessário optar por procedimentos de pré-processamento antes da aplicação das etapas subsequentes. No presente trabalho, foi escolhido descartar as colunas de nomes dos indivíduos, *ID* do passageiro, código do *Ticket*, e código da cabine, pois julgou-se que tais informações seriam pouco relevantes para as classificações. Ademais, as variáveis não numéricas restantes foram codificadas segundo a atribuição de números a cada valor único presente, na ordem em que são detectados em cada coluna ao longo das linhas. Finalmente, preencheram-se os valores faltantes com a média dos valores presentes em cada coluna.

Por outro lado, a codificação das classes "sobrevivente" $\equiv 1$ e "não-sobrevivente" $\equiv 0$ é pré-determinada pela competição na plataforma *Kaggle*, portanto o conjunto de classe é $\mathscr{C} = \{0, 1\}$.

O procedimento descrito foi realizado de maneira exploratória sobre o conjunto de treino inicialmente, para então ser unificado em uma função que realiza precisamente o mesmo pré-processamento, para ser aplicada sobre o conjunto de testes posteriormente e ser mantida a coerência entre as codificações.

Após o pré-processamento, o modelamento consiste da escolha de uma lista de p transformações $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ sobre as colunas originais do conjunto de dados, com o objetivo de aumentar a acurácia da classificação, procedimento que é conhecido como a técnica de *feature engineering* [2].

A efetividade dessa técnica se baseia em aumentar a dimensão do espaço coluna da matriz original ou transformá-lo em outro espaço que melhor represente a correlação entre as observações e as classificações para cada indivíduo, de modo que o ajuste pelo MMQ obtenha um erro menor e, consequentemente, o modelo possa prever com maior acurácia tais classificações. Por outro lado, a escolha dessas transformações é particular de cada problema e, no caso do problema tratado no presente trabalho, não se supõe que existam escolhas evidentes de tais transformações derivadas de conhecimento do domínio a que o problema pertence.

Portanto, optou-se pela proposição de 5 diferentes listas de transformações, baseadas em funções polinomiais, racionais, trigonométricas, exponenciais e outras funções não lineares, para que a mais adequada dentre elas seja escolhida programaticamente segundo uma métrica de performance.

Como o modelo não deve ser ajustado, mesmo parcialmente, ao conjunto de testes usado para obtenção de pontuação na plataforma *Kaggle*, pois isso configuraria *Data Leakage i.e.* uso de informações proibidas sobre os alvos de classificação de forma a tornar o problema consideravelmente mais fácil, a métrica de performance escolhida se baseia na subdivisão aleatória em proporção fixa do conjunto de treino total em dois subconjuntos: um sobre o qual o modelo é ajustado, também chamado conjunto de treino, e outro para validação, de modo que sejam disjuntos e, unidos, resultem no conjunto de treino original.

Dessa forma, para cada lista de transformações, por 20 iterações, a subdivisão aleatória do conjunto de treino é realizada 10 vezes em cada iteração, escolhendo-se o ajuste, dentre os 10 gerados a partir de cada subdivisão, que resulta na máxima acurácia da classificação do subconjunto de validação, e, então, é calculada a média ao longo dessas 20 iterações dessas máximas acurácias. O cálculo de valores médios para cada lista de transformações é necessário, pois, como a subdivisão do conjunto de treino é aleatória, há variações das acurácias obtidas na validação.

Os valores de acurácias médias em uma execução do código são apresentados na Tabela 1, em que a coluna Tipo indica a forma das transformações.

Tabela 1: Acurácias médias por lista de transformações em uma execução.

Lista de transformações	Tipo	Acurácia média
1	Linear	73,92 %
2	Polinomial	75,79 %
3	Trigonométrica ou Exponencial	74,93 %
4	Racional	77,23 %
5	Sinais Alternados	73,65 %

As listas de 1 a 5 em 1 são descritas em maior detalhamento nos comentários do código, no entanto, destacam-se as listas 1 e 4. A primeira corresponde a usar apenas a identidade como função transformadora *i.e.* o ajuste do modelo sobre a matriz de observações transformada corresponde a um simples ajuste linear sobre as colunas originais. Por outro lado, a lista 4 corresponde a funções da forma (11), que são aplicadas elemento-a-elemento sobre cada uma das colunas originais do conjunto de dados de treino.

$$f_k(x) = \frac{1}{(|x|+1)^k}. \; \forall k \in \{1,2,3\} \tag{11}$$

Com isso, observa-se um ganho de $3,31\%$ de acurácia pela lista 4 em relação à lista 1, que representa o modelo sem *feature engineering*, o que ressalta a efetividade da *feature engineering* realizada.

Destaca-se também que, durante a avaliação das listas, a função de transformação $\mathscr{T}$, como definida em (2), foi escolhida segundo o método simples descrito em (10), com escolha do hiperparâmetro α por meio de um método heurístico.

O método heurístico escolhido consiste em, após realizar o ajuste por mínimos quadrados dos parâmetros θ_i da função $\tilde{f}$ como descrita em (1), tomar a média simples entre as médias ao longo dos valores de saída da função $\tilde{f}$ para os conjuntos de dados de cada uma das classes. Com isso, obtém-se um valor de α que considera desvios dos valores de saída de $\tilde{f}$ em relação aos valores de $\mathscr{C}$, o que permite uma melhor separação entre os valores de saída para indivíduos de classes diferentes do que se fosse utilizado $\alpha = (0+1)/2 = 0.5$.

Como verificação disso, foi realizado o mesmo procedimento de se calcular a acurácia média realizado com as listas de transformações, considerando a lista 1 e o hiperparâmetro α fixo em 0.5, obtendo-se acurácia média de apenas $70,61\%$, o que demonstra a significância da escolha de α.

Então, após a determinação da melhor lista de transformações para *feature engineering*, a saber, a lista 4, o ajuste do modelo foi realizado através do MMQ com decomposição SVD reduzida (como descrito em §2). Disso resulta uma lista de parâmetros do ajuste, cada um correspondente a uma coluna transformada, a serem utilizados em predições subsequentes realizadas pelo modelo.

Destaca-se que o ajuste escolhido é aquele que resulta na maior acurácia sobre o conjunto de validação, assim como na etapa de escolha da *feature engineering*, no entanto, para obtenção de maior acurácia, as subdivisões aleatórias são realizadas por 500 vezes.

Com isso, mediu-se a acurácia das classificações desse modelo sobre o conjunto de dados de teste, obtendo-se $77,75\%$, e foi calculada a matriz de confusão, representada na Tabela 2.

Tabela 2: Matriz de confusão.

	Predição		
Saída	$\hat{y}=0$	$\hat{y}=1$	Total
$y=0$	215	45	260
$y=1$	48	110	158
$y=\{0;1\}$	263	155	418

Da matriz de confusão em 2, obtiveram-se as taxas de falsos positivos por classe, apresentadas na Tabela 3.

Tabela 3: Taxas de falsos positivos por classe.

Classe	Taxas de falsos positivos
0	30,38 %
1	17,31 %

As taxas em 3 mostram uma tendência do modelo classificador de tomar indivíduos da classe 1, como indivíduos da classe 0, pois a taxa de falsos positivos para a classe 0 é $30,38/17,31 \approx 1.76$ vezes maior do que para a classe 1.

Isso indica um modelo enviesado, portanto foram considerados os números de observações por classe presentes no conjunto de treino, e verificou-se que há cerca de 60% mais exemplos da classe 0 do que da classe 1. Esse desbalanceamento entre as classes é notadamente algo que pode causar enviesamento do modelo [6].

Logo, tenta-se reduzir esse viés através do ajuste fino do hiperparâmetro α. Para isso, realiza-se uma busca em um intervalo (busca em grade) pelo valor de α que gera a máxima acurácia de classificação sobre um dado conjunto de treino. Além disso, o intervalo considerado para as buscas é definido em torno do valor de α determinado pela heurística discutida anteriormente.

Com isso, foi possível alcançar uma acurácia de 78% sobre o conjunto de teste, mesmo assim, ao longo de diversas execuções do código, as acurácias dos modelos com e sem busca se mostram similares e, assim como mostra o exemplo de matriz de confusão em 4, o modelo com a busca pelo valor ótimo de α não elimina a discrepância entre as taxas de falsos positivos, que permanecem similares às do modelo sem busca.

Tabela 4: Matriz de confusão do modelo com otimização do α.

	Predição		
Saída	$\hat{y} = 0$	$\hat{y} = 1$	Total
$y = 0$	218	42	260
$y = 1$	50	108	158
$y = \{0; 1\}$	268	150	418

Finalmente, para comparar a implementação realizada com a de uma biblioteca de ajustes, foi gerado modelo classificador por meio das utilidades de regressão linear disponíveis na biblioteca `LsqFit.jl`. De forma que as diferenças entre os modelos criados através da implementação feita e aqueles criados através das utilidades da biblioteca são apenas duas, a saber:

1. A solução do problema de mínimos quadrados associado ao problema de classificação e

2. A seleção do conjunto de treino

A solução do problema de mínimos quadrados realizada através da biblioteca aplica uma técnica numérica genérica detalhada na documentação. Por outro lado, o modelo gerado através dos parâmetros da regressão feita pelas utilidades da biblioteca corresponde a um ajuste ao conjunto de treino completo *i.e.* sem as subdivisões aleatórias.

Desse modo, calculou-se a acurácia das classificações feitas por esse modelo sobre o conjunto de testes e obteve-se o valor de $75,84\%$. Essa acurácia é considerada comparável à obtida com a implementação feita, portanto, considera-se a solução do problema de classificação obtida satisfatória.

4 Conclusões

Verifica-se neste trabalho a possibilidade de se utilizar um classificador baseado no método dos mínimos quadrados para se obter acurácia de pelo menos 78% no problema proposto, performance essa bastante inferior se comparada à de um classificador baseado em redes neurais como em [1], o que é esperado pela simplicidade do método e pela complexidade do problema geral de classificação. Ademais, tal complexidade deve ser o fator mais significativo para explicar as diferenças pequenas de acurácia nas diferentes funções testadas no processo de *feature engineering* e no procedimento de otimização do hiperparâmetro α.

Por fim, pode-se afirmar que o trabalho atingiu seus objetivos, pois o problema de classificação foi resolvido com acurácia considerada satisfatória, e a comparação dos resultados da implementação realizada com aqueles da implementação disponível através da biblioteca `LsqFit.jl` evidenciou performances similares.

Referências

[1] Barhoom, A. M. et al. Predicting Titanic Survivors using Artificial Neural Network, *International Journal of Academic Engineering Research (IJAER)*, pages 8-12, volume 3, issue 9, 2019. ISSN: 2643-9085

[2] Boyd, S. and Vandenberghe, L. Introduction to Applied Linear Algebra: *Vectors, Matrices, and Least Squares, 1a. edição.* Cambridge University Press, Cambridge, 2018.

[3] Golub, G. H. and Van Loan, C. F. *Matrix Computations, 3a edição.* The Johns Hopkins Univesity Press, Baltimore, 2013.

[4] Feng, Jiashi, et al. Robust logistic regression and classification. In *Advances in neural information processing systems*, pages 253-261, 2014.

[5] Jeatrakul, P. and Wong, K. W. Comparing the performance of different neural networks for binary classification problems, *Eighth International Symposium on Natural Language Processing, Bangkok*, pages 111-115, Thailand, 2009. DOI: 10.1109/SNLP.2009.5340935

[6] Hand, D. J. and Vinciotti, V. Choosing k for two-class nearest neighbour classifiers with unbalanced classes, *Pattern Recognition Letters*, Pages 1555-1562, Volume 24, Issues 9–10, 2003. ISSN: 0167-8655