Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 10, n. 1, 2023.

Trabalho apresentado no XLII CNMAC, Universidade Federal de Mato Grosso do Sul - Bonito - MS, 2023

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Um classificador baseado em programação por metas para triagem de COVID-19 considerando sintomas

Ricardo Soares Oliveira[1]
IMECC/UNICAMP, Campinas, SP e IFG, Itumbiara, GO
Rodolfo de Carvalho Pacagnella[2]
FCM/UNICAMP, Campinas, SP
Washington A. Oliveira[3], Cristiano Torezzan[4]
FCA/UNICAMP, Limeira, SP

Resumo. A categorização de dados em classes é uma tarefa comum em diversas áreas e também uma das principais aplicações de aprendizado de máquina (AM). Enquanto a maioria dos métodos de AM utiliza uma abordagem estatística, modelos de classificação baseados em programação matemática surgem como alternativas. Este trabalho investiga a eficácia de modelos inspirados em programação por metas para classificação supervisionada, tendo como motivação um problema de triagem de COVID-19 com base nos sintomas. Os resultados obtidos permitiram estratificar a previsão em categorias de decisão, além disso, os modelos testados responderam de forma satisfatória quando comparados a modelos clássicos, como a regressão logística e máquina de vetores suporte.

Palavras-chave. Classificação, Aprendizado de Máquina, Programação por Metas, COVID-19.

1 Introdução

A pandemia da COVID-19 teve um grande impacto global, resultando em esforços significativos de pesquisa para compreender a doença, reduzir sua disseminação e mitigar seus efeitos sociais e econômicos. Alguns países, como a Coreia do Sul, se destacaram na implementação de estratégias eficazes de controle da pandemia, como "testar, rastrear e isolar", reduzindo significativamente o número de casos [2]. Enquanto a maioria dos métodos de Aprendizado de Máquina (AM) utiliza uma abordagem estatística, modelos de classificação baseados em programação matemática surgem como alternativas promissoras. A programação por metas, por exemplo, tem sido utilizada para modelar problemas de classificação supervisionada [3].

O estudo utiliza uma base de dados reais anonimizada, obtida junto ao Centro de Atenção Integral à Saúde da Mulher – CAISM - UNICAMP, no âmbito de estudos realizados junto ao *Brazilian Institute of Data Science* (BI0S) , um Centro de Pesquisa Aplicada em Inteligência Artificial financiado pela FAPESP (20/09838-0). Os dados contemplam um total de 1.102 pacientes gestantes, que reportaram 13 sintomas associados a COVID e foram submetidas a testes RT-PCR, cujos resultados estão incluídos na referida base.

Para fins de comparação, também foram implementados os modelos clássicos de regressão logística [5] e máquina de vetores suporte (*"Support Vector Machine"*(SVM)) [8], uma vez que eles têm abordagens metodológicas comparáveis à programação por metas. A regressão logística é um dos modelos mais comuns para a tarefa de classificação, um método probabilístico que visa

[1]r179370@dac.unicamp.br
[2]rodolfop@unicamp.br
[3]waoliv@unicamp.br
[4]torezzan@unicamp.br

estimar o valor de classe de uma variável dependente à partir de outras variáveis discretas e contínuas [5]. SVM é um classificador de aprendizado supervisionado que engloba ambos os tipos binário e multi-classes [8].

Os resultados obtidos permitiram estratificar a previsão em categorias de decisão, além disso, os modelos testados responderam de forma satisfatória quando comparados a modelos clássicos, como a regressão logística e máquina de vetores suporte.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta uma breve revisão da metodologia de programação por metas; a Seção 3 descreve os dois modelos de programação por metas estudados que permitem realizar classificação de dados; o estudo de caso é descrito na Seção 4 e resultados são apresentados e discutidos na Seção 5. Por fim, a Seção 6 apresenta as conclusões do trabalho.

2 Programação por metas

Modelos com múltiplos objetivos conflitantes têm sido estudados com sucesso no campo da Pesquisa Operacional considerando a metodologia de programação por metas (do inglês *goal programming*), a qual foi proposta pela primeira vez em [1] como uma adaptação da programação linear em um contexto de compensação entre os objetivos. Sejam m funções objetivo reais com variáveis de decisão n-dimensional $(f_i(x), i = 1, \ldots, m)$ e suponha que um tomador de decisão esteja interessado em atingir simultaneamente m alvos numéricos $t_1, \ldots, t_m$, no sentido de que o i-ésimo objetivo se aproxima do i-ésimo alvo [3] obedecendo à equação $f_i(x) + n_i - p_i = t_i$, onde n_i e p_i são variáveis de decisão não negativas conhecidas como desvios negativo e positivo em relação ao alvo t_i, respectivamente. Note que $n_i > 0$ quantifica o quanto falta para o objetivo atingir o alvo, enquanto $p_i > 0$ quantifica o quanto o alvo foi ultrapassado pelo objetivo. Portanto, ao minimizar o objetivo em relação ao alvo é indesejável obter $p_i > 0$, enquanto que ao maximizar o objetivo em relação ao alvo é indesejável obter $n_i > 0$. No entanto, se o objetivo precisa atingir o alvo na igualdade, é indesejável obter ambos $p_i > 0$ e $n_i > 0$.O seguinte modelo matemático de programação por metas representa o caso geral em que todas as funções objetivos são de minimização [3].

$$\begin{aligned}
\min \quad & z = h(n, \underline{p}) && (1)\\
\text{s.a} \quad & f_i(x) + n_i - \underline{p_i} = t_i, \quad i = 1, \ldots, m, && (2)\\
& x \in F, && (3)\\
& n_i, p_i \geq 0, \quad i = 1, \ldots, m, && (4)
\end{aligned}$$

onde a função de ativamento $z = h(n, \underline{p})$ é uma combinação (linear ou não linear) das entradas dos vetores de variáveis de desvios $n = (n_1, \ldots, n_m)^T$ e $p = (p_1, \ldots, p_m)^T$, $f_i(x)$ é um dos objetivos conflitantes que tem o valor alvo t_i para ser alcançado e F é chamado de conjunto de restrições rígidas para a variável x, no sentido que (2) é uma restrição branda devido às variáveis de desvios n_i and p_i. Os m objetivos são definidos no sentido de que "menor é melhor", então somente as variáveis indesejáveis de minimização precisam ser penalizadas em z. Essas variáveis aparecem sublinhadas em (1) e (2) para destacar. T. Mehrdad [7] classificaram os modelos de programação por metas nas abordagens ponderado, lexicográfico, MinMax (*Tchebychev*) e estendido.

3 Classificação baseada em programação por metas

As primeiras ideias na direção de modelos de classificação baseados em programação matemática foram propostas em [4] e em [3], onde os autores propõem modelos de classificação binária. inspirada em programação por metas que permitem a introdução de diferentes limiares de decisão. Os modelos a seguir são baseados nas ideias apresentadas em [3].

DOI: 10.5540/03.2023.010.01.0108

3.1 Modelo 1

O Modelo 1, proposto por [3], considera um problema de classificação binária, com m variáveis explicativas (atributos), cujo conjunto de dados possui n_1 observações do Tipo-A e n_2 do Tipo-B. Seja a_{ij} o valor da $j-$ésima variável explicativa associado à $i-$ésima observação do Tipo-A e b_{ij} o valor da $j-$ésima variável explicativa da $i-$ésima observação do Tipo-B. São determinadas $(m+1)$ variáveis de decisão, $\{x_0, x_1, \ldots, x_m\}$, de modo a minimizar erros de classificação, apresentado a seguir:

$$\min \quad \sum_{i=1}^{n_1} \left(n_i^{(a)}\right) + \sum_{i=1}^{n_2} \left(p_i^{(b)}\right) \tag{5}$$

$$\text{s.a} \quad \sum_{j=1}^{m} a_{ij}x_j + n_i^{(a)} - p_i^{(a)} = x_0, \quad i = 1, \ldots, n_1, \tag{6}$$

$$\sum_{j=1}^{m} b_{ij}x_j + n_i^{(b)} - p_i^{(b)} = x_0, \quad i = 1, \ldots, n_2, \tag{7}$$

$$\sum_{j=1}^{m} x_j = 1, \tag{8}$$

$$-\alpha \leq x_j \leq \alpha, \quad j = 1, \ldots, m. \tag{9}$$

Neste modelo, as $(m+1)$ variáveis de decisões definem os coeficientes do hiperplano $x_1y_1 + x_2y_2 + \cdots + x_my_m = x_0$, onde y é um vetor m dimensional que representa uma observação no espaço dos atributos. O parâmetro α é definido previamente, de modo que $\alpha > \max_j |x_j|$, e pode ser utilizado para calibrar a especificidade e a sensibilidade do método. Para uma classificação perfeita, todas as observações do tipo-A deveriam estar do lado positivo do hiperplano e todas do tipo-B deveriam estar do lado negativo.

Na prática, em função de características numéricas, o Modelo 1 permite que várias observações caiam exatamente sobre o hiperplano, não permitindo sua classificação. Além disso, amostras que fiquem muito próximas do hiperplano separador introduzem sensibilidade ao modelo. Uma solução para isto é criar faixas de classificação, como proposto no Modelo 2, apresentado a seguir.

DOI: 10.5540/03.2023.010.01.0108

3.2 Modelo 2

O Modelo 2, proposto por [3], visando reduzir erros de classificação na região do hiperplano separador, cria uma zona de separação de tamanho 2β, através da seguinte formulação:

$$\min \quad \sum_{i=1}^{n_1} \left(n_i^{(a)}\right) + \sum_{i=1}^{n_2} \left(p_i^{(b)}\right) \tag{10}$$

$$\text{s.a} \quad \sum_{j=1}^{m} a_{ij}x_j + n_i^{(a)} - p_i^{(a)} = x_0 + \beta, \quad i = 1, \ldots, n_1, \tag{11}$$

$$\sum_{j=1}^{m} b_{ij}x_j + n_i^{(b)} - p_i^{(b)} = x_0 - \beta, \quad i = 1, \ldots, n_2, \tag{12}$$

$$\sum_{j=1}^{m} x_j = 1, \tag{13}$$

$$-\alpha \leq x_j \leq \alpha, \quad j = 1, \ldots, m. \tag{14}$$

$$\tag{15}$$

Nesta formulação, o parâmetro β controla o tamanho de uma faixa de decisão em torno do hiperplano separador, que permite criar duas novas categorias de decisão, que podem estar associadas a graus diferentes de certeza na classificação. A Figura 1 ilustra os dois modelos apresentados, para um problema com dois atributos. A linha sólida x_0 ilustra o limiar de classificação do Modelo 1 e às duas linhas pontilhadas representam os limiares das faixas do Modelo 2. Nesta figura, pontos azuis deveriam estar todos abaixo do hiperplano e pontos azuis acima. Assim, pontos que estão em lados opostos do esperado indicam erros de classificação.

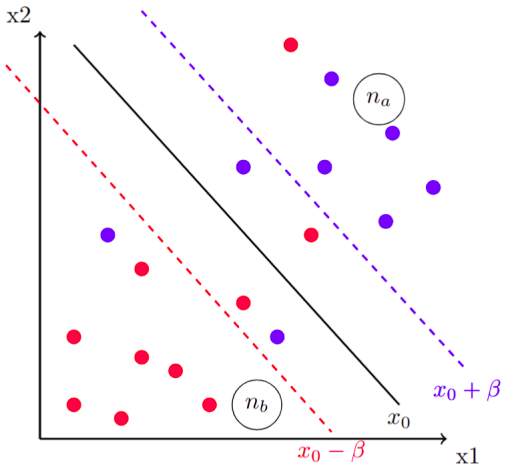

Figura 1: Modelo básico de classificação. Fonte: adaptado [6]

Para fins de classificação, as quatro faixas definidas pelo Modelo 2, representam os seguintes níveis de classificação: "definitivamente A", "provavelmente A", "provavelmente B", "definitivamente B". Pontos que ficarem sobre o hiperplano de nível x_0 serão declarados como "indefinidos".

DOI: 10.5540/03.2023.010.01.0108

4 Estudo de caso: triagem de COVID-19 com base em sintomas

Para investigar a viabilidade e acurácia dos Modelos 1 e 2 para classificação de casos de COVID-19 foram realizados testes computacionais empíricos, com base em dados reais anonimizados, contendo o resultado de testes RT-PCR de mulheres gestantes com suspeita de COVID-19. Os dados foram obtidos pelo Centro de Atenção Integral à Saúde da Mulher – CAISM, no âmbito de projetos desenvolvido em conjunto com *Brazilian Institute of Data Science* (BI0S).

A Figura 2 apresenta cinco linhas de ($m = 1102$) referentes aos dados clínicos das pacientes e as ($n = 14$) colunas, sendo treze os sintomas e a décima quarta o resultado RT-PCR de COVID-19, onde (positivo= 1) e (negativo= 0). O objetivo do trabalho é aplicar os modelos para classificar o mais adequado possível o paciente, de acordo com os sintomas. Para isso é feito um treinamento com 70% dos dados e a validação com 30% dos dados restantes. O conjunto de treino foi dividido em dois Tipos: A e B. Para serem inseridos nos parâmetros a_{ji} e b_{ij} respectivamente. A separação se da através da análise supervisionada da última coluna, COVID.

Row	Disfalgia Int64	Febre38 Int64	Tosse Int64	Congestao_nasal Int64	Dor_de Cabeça Int64	Mialgia Int64	Dispineia Int64	Vomito Int64	Diarreia Int64	hiposia Int64	Perda do paladar Int64	Calafrios Int64	Oxigenio95 Int64	COVID Int64
1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Figura 2: Cabeçalho e perfil do conjunto de dados utilizado, com destaque para as 5 primeiras amostras.

O conjunto de dados não é balanceado com relação ao resultado do teste para COVID-19, contendo $35,11\%$ de amostras positivas e $64,11\%$ negativas. Por isso, cada método foi testado em dois cenários: sem balanceamento, denotado por (sb) e com balanceamento, denotado por (cb). O balanceamento foi realizado utilizando-se a metodologia de subamostragem (*undersample*) aleatória da classe predominante, de modo a resultar num subconjunto das amostras contendo o mesmo percentual de casos positivos e negativos.

4.1 Testes Computacionais

O Modelo 1 foi testado em quatro cenários adicionais denominados A, B, C e D, que consideram $\alpha = 1$ e diferentes intervalos para a variável x_j, como segue: (A) $0 \leq x_j \leq 1$; (B) $|x_j| \leq \alpha$; (C) x_j livre; e (D) $|x_j| \leq \alpha$, com a eliminação da Restrição (8). No Modelo 2 a variação é realizada no conjunto de valores do parâmetro $\beta = \{0.1, 0.5, 1.0, 2.0\}$. Em ambos os modelos, os testes foram realizados para os casos sem balanceamento (sb) e com balanceamento (cb).

Para avaliar a qualidade dos resultados, foram utilizadas métricas usuais em AM para modelos de classificação. A acurácia, que é a razão entre o número de amostras classificadas corretamente e o total de amostras. Além disso, para aplicações em saúde é relevante controlar as taxas de Falso Negativo e de Falso Positivo. Para isso, também foram analisadas as taxas de acerto dos pacientes com classificação Positivo e Negativa predito pelo modelo. Conhecidas como taxa de verdadeiro positivo (ou sensibilidade), em inglês (*True Postive Ratio* (TPR)), e taxa de verdadeiro negativo (especificidade)(*True Negative Ratio*(TNR)). Adicionalmente, a métrica de *f1-score*, que é a média ponderada da precisão com a *recall* também foi utilizada.

Tabela 1: Métricas de desempenho dos métodos.

Modelo	Classes	% Acerto	% Erro	Acurácia	f1-score
M1 (sb)	P	**0,61**	0,28	0,68	**0,58**
	N	0,72	0,38		
M2 (sb)	Def P	**0,78**	0,22	**0,72**	0,48
	Pro P	**0,84**	0,16		
	Indef	0,64	0,36		
	Pro N	0,60	0,40		
	Def N	0,74	0,26		
Logit (sb)	P	0,43	0,57	0,71	0,52
	N	**0,87**	0,13		
SVM (sb)	P	0,24	0,76	0,64	0,33
	N	**0,87**	0,13		
M1 (cb)	P	0,64	0,28	0,69	0,60
	N	0,71	0,35		
M2 (cb)	Def P	0,82	0,18	0,73	0,63
	Pro P	0,64	0,36		
	Indef	0,50	0,50		
	Pro N	0,75	0,25		
	Def N	**0,79**	0,21		
Logit (cb)	P	0,75	0,25	0,82	0,81
	N	**0,90**	0,10		
SVM (cb)	P	**0,99**	0,01	**0,84**	**0,82**
	N	0,70	0,30		

5 Resultados e Discussões

A Figura 3 apresenta um comparativo do desempenho dos Modelos 1 e 2 para os cenários de testes. Pode-se notar que, de forma geral, o Modelo 2 apresenta melhores resultados do que o Modelo 1.

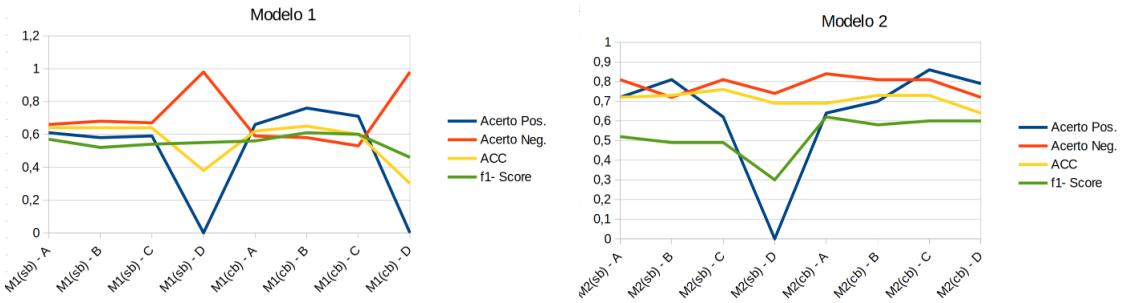

Figura 3: Resultados Computacionais dos Modelos M1 e M2

A Tabela 1 apresenta uma comparação entre as métricas de desempenho dos Modelos 1 e 2 com os métodos de regressão logística (Logit) e máquinas de suporte de vetores (SVM), considerando os problemas sem balanceamento (sb) e com balanceamento (cb).

Pode-se notar que os Modelos 1 e 2 são menos afetados pelo desbalanceamento das classes, superando os métodos clássicos Logit e SVM neste cenário. Com o balanceamento dos dados (cb), pode-se notar que todos os métodos melhoraram seus desempenhos, com destaque para a regressão logística e para o SVM que superaram os resultados dos Modelos 1 e 2, embora o Modelo 2 tenha se mantido competitivo. Isso pode evidenciar que os modelos 1 e 2 são mais aptos para dados sem balanceamento.

A análise dos percentuais de acerto nas classes definitivamente positivo (Def P), possivelmente positivo (Pro P), definitivamente negativo (Def N) e possivelmente negativo (Pro N) mostram que a medida que as amostras se distanciam do hiperplano separador as taxas de acerto aumentam -

com exceção do Def P e Pro P do Modelo 2 sem balanceamento.

De forma geral, os testes computacionais realizados neste trabalho permitem concluir que modelos de Programação Matemática podem ser boas alternativas quando se trata de problemas de classificação, sobretudo em cenários com classes desbalanceadas. Não obstante, nota-se que tanto os modelos de programação matemática (M1 e M2), quanto os de aprendizado de máquina clássico (logit e SVM) tiveram um desempenho melhor após o balanceamento (CB) dos dados.

6 Considerações Finais

Pelos resultados apresentados na Tabela 1 e a discussão apresentada na Seção 5, verifica-se que os modelos de programação por metas permitem formulações alternativas para problemas de classificação binária, cujos resultados são competitivos quando comparados com modelos clássicos de aprendizado de máquina, como a regressão logística e a máquina de suporte de vetores, sobretudo em cenários com classes desbalanceadas. Além disso, modelos de programação por metas apresentam a vantagem de estabelecer faixas de classificação, permitindo a estratificação dos resultados em níveis distintos de certeza. Investigações adicionais e trabalhos futuros podem envolver novas aplicações em saúde, bem como a introdução de mais faixas de decisão e estudos para ajuste dos parâmetros dos modelos visando melhorar o compromisso entre sensibilidade e especificidade.

Agradecimentos

FAPESP (processo 2020/09838-0) e CNPq (processo 309925/2021-5).

Referências

[1] R. O. Ferguson A. Charnes W. W. Cooper. "Optimal estimation of executive compensation by linear programming". Em: **Management science** 2 (1995), pp. 138–151. DOI: 10.1287/mnsc.1.2.138.

[2] WHO et al. **Coronavirus disease (COVID-19), 21 September 2020**. Online. Acessado em 25/03/2023, https://apps.who.int/.

[3] A. Collins E C. Hand D. F. Jones. "A classification model based on goal programming with non-standard preference functions with application to the prediction of cinema-going behaviour". Em: **European Journal of Operational Research** 177.1 (2007), pp. 515–524.

[4] N. Freede F. Glover. "Simple but powerful goal programming models for discriminant problems". Em: **European Journal of Operational Research** 7.1 (1981), pp. 44–60. DOI: 10.1016/0377-2217(81)90048-5.

[5] S. Menard. **Applied Logistic Regression Analsis**. 2a. ed. Houston: SAGE Publications, Inc, 2002. ISBN: 10.4135/9781412983433.

[6] A. I. Quezada. "Using Optimization Models to Achieve Solutions in Classification and Clustering Technique". Dissertação de mestrado. Unicamp, 2020.

[7] J. Dylan e T. Mehrdad. **Practical goal programming**. Vol. 141. Springer, 2010. ISBN: 978-1441957702.

[8] V. N. VAPNIK. **The nature os statiscal learnning theory**. Springer Briefs in Mathematics. Springer, 1999. ISBN: 0-387-98780-0.

