Aplicação da Metaheurística PSO na Identificação de Pontos Influentes por meio da Função de Sensibilidade de Casos

Adriana Aparecida Batista Costa
Centro Federal de Educação Tecnológica de Minas Gerais, PPGMMC
Av. Amazonas, 7675, 30.510-000, Belo Horizonte, MG
E-mail: adrianab@dppg.cefetmg.br

Elenice Biazi
Centro Federal de Educação Tecnológica de Minas Gerais, PPGMMC
Av. Amazonas, 7675, 30.510-000, Belo Horizonte, MG
E-mail: elenice@dppg.cefetmg.br

João Francisco de Almeida Vitor
Centro Federal de Educação Tecnológica de Minas Gerais, PPGMMC
Av. Amazonas, 7675, 30.510-000, Belo Horizonte, MG
E-mail: joaofrancisco@dppg.cefetmg.br

Resumo*: Neste trabalho é aplicada a metaheurística otimização por enxame de partículas (PSO) na identificação de pontos influentes em modelos de regressão. Foi utilizada, como função objetivo, a função de sensibilidade de casos* $g_{Cook}(\varepsilon)$ *que tem característica multimodal. A metaheurística PSO, por sua característica evolutiva, tem sido adequada no tratamento de problemas multimodais. A eficiência da metodologia proposta foi testada em conjuntos de dados simulados. Os resultados obtidos mostram que esta metodologia apresenta soluções satisfatórias na identificação de pontos influentes.*

Palavras- chave*: Modelos de regressão, pontos influentes, função de sensibilidade de casos, metaheurística PSO.*

1. Introdução

Os modelos estatísticos, geralmente, são descrições aproximadas de processos bastante complexos, que conseqüentemente podem levar a resultados imprecisos, surge então uma importante motivação para o estudo de técnicas que avaliem essa imprecisão [4].

A partir da década de 70 surgiram várias propostas relacionadas com a influência das observações nas estimativas dos coeficientes do modelo de regressão linear. As medidas de omissão de pontos foram propostas por Cook [5] e Belsley et al [2]. Um problema que pode ocorrer com a omissão individual de pontos é o que se denomina "mascaramento", ou seja, deixar de detectar pontos conjuntamente influentes. A função de influência é abordada nos trabalhos de Hampel [13] e Cook [6], onde são analisadas perturbações em probabilidades associadas a casos. Uma proposta inovadora na análise de diagnóstico em regressão, denominada influência local, foi apresentada por Cook [7], que propõe avaliar a influência conjunta das observações sob pequenas perturbações no modelo. Uma nova medida de influência, baseada no quadrado da norma do vetor de previsões, foi proposta por Peña [15].

Algumas aplicações da metaheurística algoritmo genético, utilizando métodos combinatórios para detectar *outliers* em modelos de regressão, podem ser encontradas em [8] e [18]. Em [19] foi utilizada a metaheurística PSO para detectar *outliers*, estudando o comportamento das projeções dos conjuntos de dados.

A função de sensibilidade de casos (*Case Sensitivity Function*) é uma nova abordagem para análise de dados influentes. Esta função foi proposta por Critchley [9] e explorada em Biazi [3] e Critchley et al. [10]. Esta metodologia mostrou-se eficiente para identificar pontos influentes de uma forma geral, inclusive para múltiplos *outliers* em problemas com dados

multivariados. Entretanto, nestes trabalhos, devido às limitações computacionais, esta metodologia foi testada em pequenos conjuntos com até 40 dados.

No presente trabalho foi verificada a eficiência da função de sensibilidade de casos para identificar pontos influentes, empregando a metaheurística PSO, em conjuntos simulados com 40, 200 e 500 dados, obtendo resultados satisfatórios. O conhecimento de tais pontos influentes contribui para uma modelagem estatística mais eficiente já que os pontos influentes individualmente, ou em conjunto, podem produzir grandes alterações em aspectos importantes da análise ou contribuir para o fornecimento de resultados imprecisos.

2. Função de Sensibilidade de Casos (*Case Sensitivity Function*)

A função de sensibilidade de casos permite a identificação de múltiplos pontos influentes, mesmo na presença de "mascaramento". Esta função é uma extensão da função de influência de Hampel [13], representada por $g_T(\boldsymbol{\varepsilon})$, onde ε é um vetor de perturbações e T é um funcional estatístico.

A forma geral da função $g_T(\boldsymbol{\varepsilon})$ é dada por**:**

$$g_T(\varepsilon) = T.[(1-\sum_{i=1}^{n}\varepsilon_i)\hat{F} + \sum_{i=1}^{n}\varepsilon_i\delta_{x_i}] = T.[\sum_{i=1}^{n} p_i(\varepsilon)\delta_{x_i}] \quad (1)$$

sendo $p_i(\varepsilon) = \frac{1}{n} + \varepsilon_i + \bar{\varepsilon}$, δ_i função de densidade pondo massa 1 em x ∈ $\Re^n$ e $\hat{F}$ função de densidade empírica de uma amostra aleatória $x_1, x_2, ..., x_p$.

Para avaliar a influência conjunta é necessário explorar o efeito de uma pequena perturbação ε no modelo, comparando $g_T(\varepsilon)$ com $g_T(0)$, o valor de g_T sem perturbação. O objetivo é maximizar $| g_T(\varepsilon) - g_T(0) |$ em uma região determinada pelas probabilidades [3]:

$$0 \le p_i \le \frac{1}{n-k} \qquad para\ \ 1 \le i \le n \quad (2)$$

sendo $\sum_{i=1}^{n} p_i = \frac{1}{n} + \varepsilon_i + \bar{\varepsilon}$, onde n é o número de observações e k é o número de observações influentes.

Neste trabalho é utilizada, como função objetivo, a função de sensibilidade de casos para a distância de Cook. A função $g_T(\varepsilon)$ para a distância de Cook é expressa por :

$$g_{Cook}(\varepsilon) = \frac{(g_{\hat{\beta}}(\varepsilon) - g_{\hat{\beta}}(0))' X' X (g_{\hat{\beta}}(\varepsilon) - g_{\hat{\beta}}(0))}{ps^2} \quad (3)$$

sendo $(g_{\hat{\beta}}(\varepsilon) - g_{\hat{\beta}}(0)) = [(X'EX)^{-1}X'Ey - (X'X)^{-1}X'y]'$, $E = diag\left(\frac{1}{n} + \varepsilon_i + \bar{\varepsilon}\right)$, $\hat{\beta}$ o vetor de parâmetros estimados, p o número de variáveis e s^2 a variância amostral.

3. Metaheurística PSO

O método de otimização por enxame de partículas (PSO - Particle Swarm Optimization) é uma técnica de computação estocástica baseada em dinâmica de populações. Desenvolvido por Kennedy e Eberhart [14], este método consiste na otimização de uma função objetivo por meio da troca de informações entre indivíduos (partículas) de uma população (enxame).

Segundo Shi e Eberhart [17], no algoritmo PSO, cada partícula, tratada como um ponto no espaço D-dimensional, representa uma solução potencial para um problema, ajustando sua posição com base na própria experiência e na experiência do grupo. A cada iteração, a velocidade é atualizada, conforme equação (4). A nova posição da partícula é determinada pela soma da sua posição atual e a nova velocidade, de acordo com a equação (5).

$$v_{id} = w.v_{id} + c_1.r_1(pbest - x_{id}) + c_2.r_2(gbest - x_{id}) \tag{4}$$

$$x_{id} = x_{id} + v_{id} \tag{5}$$

sendo v_{id} a velocidade atual da partícula(i), w o peso inercial que equilibra a exploração global e local, c_1 e c_2 duas constantes positivas, r_1 e r_2 números aleatórios entre [0,1], *pbest* a melhor posição já alcançada pela partícula e *gbes*t a melhor posição encontrada pelo enxame.

A equação corresponde à soma de três termos distintos: o primeiro refere-se à inércia da partícula; o segundo é um termo cognitivo relativo à atração da partícula ao melhor ponto que já encontrou; e o terceiro é um termo social que representa a colaboração entre as partículas.

A atualização da velocidade de cada partícula depende de parâmetros que devem ser ajustados a cada problema a ser otimizado. Em [17] é sugerido que $c_1 = c_2 = 2$, de forma a manter o equilíbrio entre as partes cognitiva e social do comportamento da partícula. O peso inercial (w) permite a diversidade de exploração do espaço de busca. Valores altos para o peso inercial facilitam a exploração global, ao passo que valores menores favorecem a exploração local. Em [12] é sugerido que w seja escolhido no intervalo [0,7, 1,4].

A fim de melhorar a eficiência do algoritmo PSO, foi proposta por Eberhart e Shi [12] a seguinte equação para que o valor de w varie a cada iteração do algoritmo:

$$w^{it} = (w_i - w_f).\left(\frac{it_{\max} - it}{it_{\max}}\right) + w_f \tag{6}$$

sendo w_i o valor inicial para o coeficiente de inércia e w_f o valor final para o coeficiente de inércia.

Os passos para a implementação do algoritmo básico PSO são os seguintes [11]:

Passo 1: Inicializar a população de partículas com posições e velocidades aleatórias no espaço D dimensional;
Passo 2: Avaliar a aptidão de cada uma das partículas;
Passo 3: Comparar o valor obtido da partícula com *pbest.* Se o valor for melhor, atualizar *pbes*t com o novo valor;
Passo 4: Comparar o valor obtido com o melhor valor global *gbest*. Se for melhor, atualizar *gbest* com o novo valor;
Passo 5: Atualizar a velocidade da partícula de acordo com a equação(4) ;
Passo 6: Atualizar a posição da partícula de acordo com a equação(5) ;
Passo 7: Repetir os passos 2-6 até que algum critério de parada seja alcançado.

4. Implementação Computacional

A implementação do algoritmo PSO foi realizada utilizando o software Matlab versão 7.6, em um computador Core 2 Duo com 2GB de memória RAM, HD 160 GB e sistema operacional Windows XP.

Neste trabalho foram adotados os seguintes critérios de parada: o número máximo de iterações sem melhora (itsm= 0,1*itmax) ou o número máximo de iterações(itmax).

O algoritmo PSO adaptado ao problema tratado neste trabalho segue os seguintes passos:

Passo 1: Inicializar a população de partículas com posições e velocidades aleatórias, a partir das equações:
$\varepsilon_0 = \varepsilon_{min} + r_1(\varepsilon_{max} - \varepsilon_{min})$
$v_0 = \varepsilon_{min} + r_2(\varepsilon_{max} - \varepsilon_{min})$
sendo ε_{min} e ε_{max} extremos do domínio.
Passo 2: Avaliar a aptidão de cada uma das partículas de acordo com a equação:
$Fo = | g_{Cook}(\varepsilon) - g_{Cook}(0) |$;
Passo 3: Determinar a melhor posição da partícula, $pbest_i$;
Passo 4: Determinar a melhor posição do enxame, *gbest* ;
Passo 5: Se $Fo(\varepsilon_i) > Fo(pbest_i)$, atualize $pbest_i$ com a posição corrente;
Passo 6: Se $Fo(pbest_i) > Fo(gbest)$, atualize *gbest* com $pbest_i$;
Passo 7: Atualizar a velocidade de cada uma das partículas conforme equação 4;
Passo 8: Atualizar a posição de cada uma das partículas conforme equação 5;
Passo 9: Repetir os passos 2-8 até que um dos critérios de parada seja satisfeito.

5. Resultados

Para testar a metodologia proposta, foram utilizados conjuntos de dados simulados de acordo com o critério proposto por Rousseeuw [16], onde o número de dados (n) é constituído mantendo uma proporção de 0.6 pontos "bons" e 0.4 pontos "ruins", apresentando forte mascaramento. Este critério foi adotado nos trabalhos de Atkinson [1], Biazi [3], Critchely et al.[10] e Peña [15], para conjuntos com até 50 dados.

Os conjuntos de dados foram gerados de forma que, 60% dos pontos seguem o modelo $y_i = x_i + 2 + e_i$, com x_i uniformemente distribuído no intervalo [1,4] e e_i normalmente distribuído com desvio padrão igual a 0,2. Os outros 40% dos pontos são normalmente distribuídos com desvio padrão 0,5 e médias $\mu_x = 7$ e $\mu_y = 2$.

Os parâmetros do PSO adotados nos testes computacionais, que apresentaram melhor convergência, foram os seguintes: $c_1 = 1{,}95$, $c_2 = 2{,}05$, $w_i = 0{,}9$ e $w_f = 0{,}4$. O número de partículas variou de acordo com o tamanho do conjunto de dados. Desta forma, foram utilizadas 30,100 e 150 partículas para conjuntos com 40, 200 e 500 dados, respectivamente.

A tabela 1 apresenta os valores médios e os melhores resultados encontrados nas execuções do algoritmo. A coluna melhor solução corresponde ao maior valor encontrado para a função objetivo.

Nº de dados	Nº de execuções	Melhor solução	Média das soluções	Melhor tempo(s)	Tempo médio(s)	Melhor iteração	Média das iterações
40	500	$1{,}217 \times 10^4$	$1{,}2028 \times 10^4$	2,75	2,84	162	197,8
200	500	$3{,}417 \times 10^4$	$3{,}1861 \times 10^4$	51,19	58,48	701	757,2
500	50	$8{,}162 \times 10^4$	$8{,}0042 \times 10^4$	3024,2	3325,52	1751	1942,3

Tabela 1: Resultados computacionais.

Na tabela 2 estão os valores das perturbações (ε_i) obtidos para cada conjunto de dados. Em negrito são indicados os pontos influentes.

n = 40		n = 200		n = 500	
Pontos	ε_i	Pontos	ε_i	Pontos	ε_i
1 a 24	0,02	1 a 120	0,004	1 a 300	0,0016
25 a 40	**-0,025**	**121 a 200**	**-0,005**	**301 a 500**	**-0,002**

Tabela 2: Valores das perturbações (ε_i).

Os resultados apresentados na tabela 2 indicam os pesos (perturbações) associados a cada ponto quando a função objetivo $f = | g_{Cook}(\varepsilon) - g_{Cook}(0) |$ é maximizada. Observações com pesos negativos são consideradas influentes na determinação dos coeficientes do modelo de regressão. Assim, a metodologia identificou corretamente os pontos influentes associando pesos negativos para os 40% dos pontos considerados "ruins".

6. Conclusões

A metaheurística PSO mostrou-se eficiente para identificar pontos influentes em modelos de regressão, utilizando a função de sensibilidade de casos. A vantagem do uso desta metodologia está na possibilidade de tratar grandes conjuntos de dados, algumas vezes inviável por outras técnicas, em tempo computacional reduzido. Os resultados obtidos mostram que a metodologia proposta identifica corretamente os subconjuntos de dados, mesmo quando há "mascaramento" de pontos.

Referências

[1] Atkinson, R. A., "Masking unmasked", Biometrika, 73, p.533-541, 1986.

[2] Belsley, D., Kuh, E., and Welsch, R., "Regression diagnostics", John Wiley, New York, 1980.

[3] Biazi, E., "Some Aspects of Influence Analysis and a New Approach", Tese PhD, University of Warwick, 1996.

[4] Billor, N. and Loynes, R., "Local Influence: a new approach", Communs Statist, Theory Meth, 22. p.1595-1661, 1993.

[5] Cook, R., "Detection of influential observations in linear regression", Technometrics, 19, p.15-18, 1977.

[6] Cook, R. and Weisberg, S., "Residuals and influence in regression", Chapman and Hall, 1982.

[7] Cook, R., "Assessment of local influence", J.R.Statist.Assoc.,48 , p.133-169, 1986.

[8] Crawford, K. D. and Wainwright, R. L, "Applying genetic algorithms to outlier detection", Morgan Kaufmann Publishers, 1986.

[9] Critchley, F., "Discussion of leave-k-out diagnostics for time series by A.G.Bruce and R.D.Martin", J.R.Statist.Soc.,51. p.407-408,1989.

[10] Critchley, F., Atkinson, R. A., Lu, G., and Biazi, E., "Influence Analysis Basead on The Case Sensitivity Function", Royal Statistical Society, 63, Part 2, p.307-323, 2001.

[11] Eberhart, R., Simpson, P., and Dobbins, R., "Computational Intelligence PC Tools"

MA Academic Press Professional, Boston, 1996.

[12] Eberhart, R. And Shi Y., "Comparing inertia weights and constriction factors in particle swarm optimization", Proceedings *of Congress on Evolutionary Computation*, San Diego, p. 84-88, 2000.

[13] Hampel, F., " The influence curve and its role in robust estimation", J. Amer. Statist. Assoc., 69, 383-393, 1974.

[14] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization", Proc.of the IEEE, International Conference on Neural Networks, Piscataway, NJ, p. 1942-1948, 1995.

[15] Peña D., "A New Statistic for Influence in Linear Regression", Technometrics, Vol.47, nº 1, p.1-12, 2005.

[16] Rousseeauw, P.J.,Leroy, A.M., " Robust Regression and Outlier Detection",New York, Wiley, 1987.

[17] Shi, Y. and Eberhart, R., "A Modified Particle Swarm Optimizer", Proc.of the IEEE Congress on Evolutionary Computation (CEC 1998), Piscataway, NJ, p. 69-73, 1998.

[18] Tolvi, J., "Genetic algorithms for outlier detection and variable selection in linear regression models", Springer-Verlag, p. 527-533, 2004.

[19] Ye D. and Chen Z., "A New Algorithm for High-Dimensional Outlier Detection Based on Constrained Particle Swarm Intelligence" Springer Berlin, vol.5009, p.516-523, 2008.