TEMA Tend. Mat. Apl. Comput., **11**, No. 1 (2010), 41-48.

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

A.A.B. COSTA[1], E. BIAZI[2], J.F.A. VÍTOR[3], Programa de Pós-Graduação em Modelagem Matemática e Computacional, CEFET-MG, 30510-000 Belo Horizonte, MG, Brasil.

Resumo. Neste trabalho é aplicada a metaheurística Otimização por Enxame de Partículas (*Particle Swarm Optimization* -PSO) na identificação de pontos influentes. Estes pontos exercem grande influência na determinação dos coeficientes do modelo de regressão. Foi utilizada, como função objetivo, a função de sensibilidade de casos $g_{Cook}(\epsilon)$ que tem comportamento multimodal. 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, 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 técnica de 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 [14] e Cook e Weisberg [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

[1]adrianab@dppg.cefetmg.br
[2]elenice@dppg.cefetmg.br
[3]joaofrancisco@dppg.cefetmg.br

Recebido em 09 Setembro 2009; Aceito em 10 Maio 2010.

de influência, baseada no quadrado da norma do vetor de previsões, foi proposta por Peña [16].

Algumas aplicações da metaheurística algoritmo genético, utilizando métodos combinatórios para detectar *outliers* em modelos de regressão, podem ser encontradas em [9] e [20]. Em [21] 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 [10] e explorada em Biazi [3] e Critchley et al. [11]. Esta metodologia mostrou-se eficiente para identificar pontos influentes. 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. A identificação 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

Critchley [10] propôs uma nova abordagem para a função de influência de Hampel [14], denominada *Case Sensitivity Function*. A abordagem de Critchley é uma generalização da função de influência para múltiplos casos, e permite a identificação de múltiplos pontos influentes, mesmo na presença de "mascaramento". A função de sensibilidade de casos é representada por $g_T(\epsilon)$, onde ϵ é um vetor de perturbações e T é um funcional estatístico.

A forma geral da função $g_T(\epsilon)$ é dada por

$$g_T(\epsilon) = T[(1 - \sum_{i=1}^{n} \epsilon_i)\widehat{F} + \sum_{i=1}^{n} \epsilon_i \delta_{xi}], \tag{2.1}$$

onde
ϵ_i - i-ésima perturbação,
δ_{xi} - função de densidade que associa massa 1 em $x \in \Re^n$,
$\widehat{F}$ - função de densidade empírica de uma amostra aleatória $x_1, x_2, ..., x_p$.

A metodologia consiste em inserir pequenas perturbações (ϵ) no modelo e avaliar o impacto causado por estas perturbações, comparando $g_T(\epsilon)$ com $g_T(0)$. Em que $g_T(\epsilon)$ é o valor da função de sensibilidade com perturbação e $g_T(0)$ é o valor da função sem pertubação [8].

O objetivo é maximizar $|g_T(\epsilon) - g_T(0)|$ sujeita a uma região determinada pelas probabilidades:

$$0 \leq p_i \leq \frac{1}{n-k} \qquad \text{para} \quad 1 \leq i \leq n,$$

sendo $p_i = \frac{1}{n} + \epsilon_i - \bar{\epsilon}$ e $\sum_{i=1}^{n} p_i = 1$, em que n é o número de observações e k é o número de observações influentes.

O vetor de perturbações $\epsilon = (\epsilon_1, ..., \epsilon_n)'$ que maximiza a distância $|g_T(\epsilon) - g_T(0)|$ contém os pesos associados a cada ponto do conjunto de dados. Pontos com pesos negativos são considerados influentes no ajuste do modelo.

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

$$g_{Cook}(\epsilon) = \frac{(g_{\widehat{\beta}}(\epsilon) - g_{\widehat{\beta}}(0))' X' X (g_{\widehat{\beta}}(\epsilon) - g_{\widehat{\beta}}(0))}{ps^2}, \tag{2.2}$$

sendo $g_{\widehat{\beta}}(0) = (X'X)^{-1}X'y$, $g_{\widehat{\beta}}(\epsilon) = (X'EX)^{-1}X'Ey$, E = diag$(\frac{1}{n} + \epsilon_i - \bar{\epsilon})$, p o número de variáveis e s^2 a variância amostral.

3. Otimização por Enxame de Partículas

O método de otimização por enxame de partículas (*Particle Swarm Optimization - PSO*) é uma técnica de computação estocástica baseada em dinâmica de populações. Desenvolvido por Kennedy e Eberhart [15], 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 [19], 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 (3.1). A nova posição da partícula é determinada pela soma da sua posição atual e a nova velocidade, de acordo com a equação (3.2)

$$v_i^{k+1} = w.v_i^k + c_1.r_1(pbest_i - x_i^k) + c_2.r_2(gbest - x_i^k), \tag{3.1}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \tag{3.2}$$

sendo v_i a velocidade atual da partícula i, c_1 e c_2 duas constantes positivas, w o peso inercial, r_1 e r_2 números aleatórios uniformemente distribuídos entre [0,1], $pbest_i$ a melhor posição já alcançada pela partícula e $gbest$ a melhor posição encontrada pelo enxame.

A atualização da velocidade de cada partícula depende de parâmetros que devem ser ajustados a cada problema a ser otimizado. Em [19] 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 [13] é sugerido que w seja escolhido no intervalo [0,7, 1,4]. O número de partículas presentes na população é determinado empiricamente, com base na dimensionalidade e percepção e dificuldade de um problema [17].

Os passos para a implementação do algoritmo básico PSO são os seguintes [12]:

1. Inicializar a população de partículas com posições e velocidades aleatórias no espaço D dimensional;
2. Avaliar a aptidão de cada uma das partículas;
3. Comparar o valor obtido da partícula i com *pbest*. Se o valor for melhor, atualizar *pbest* com o novo valor;
4. Comparar o valor obtido com o melhor valor global *gbest*. Se for melhor, atualizar *gbest* com o novo valor;
5. Atualizar a velocidade da partícula de acordo com a equação (3.1);
6. Atualizar a posição da partícula de acordo com a equação (3.2);
7. Repetir os passos 2-6 até que algum critério de parado seja alcançado.

4. Proposta de uma Heurística PSO para a Identificação de Pontos Influentes por meio da Função de Sensibilidade de Casos

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 ($it_{sm} = 0,1 * it_{max}$) ou o número máximo de iterações (it_{max}). O número máximo de iterações foi de 200, 800 e 2000 para os conjuntos com 40, 200 e 500 dados, respectivamente. Caso o algoritmo atinja o número máximo de iterações sem melhora da solução ou o número máximo de iterações, o programa finaliza sua execução.

O algoritmo PSO, adaptado ao problema tratado neste trabalho, pode ser descrito pelos seguintes passos:

Passo 1: Inicializar a população de partículas com posições e velocidades aleatórias, a partir das equações:

$$\epsilon_0 = \epsilon_{min} + r_1(\epsilon_{max} - \epsilon_{min})$$
$$v_0 = \epsilon_{min} + r_2(\epsilon_{max} - \epsilon_{min})$$

sendo ϵ_{min} e ϵ_{max} extremos do domínio;

Passo 2: Avaliar a aptidão de cada uma das partículas de acordo com a equação:

$$Fo = |g_{Cook}(\epsilon) - g_{Cook}(0)|;$$

Passo 3: Determinar a melhor posição da partícula, $pbest_i$;

Passo 4: Determinar a melhor posição do enxame, *gbest*;

Passo 5: Se $Fo(\epsilon_i) > Fo(pbest_i)$, 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 de acordo com a equação (3.1);

Passo 8: Atualizar a posição de cada uma das partículas conforme equação (3.2);

Passo 9: Repetir os passos 2-8 até que um dos critérios de parada seja satisfeito.

5. Resultados

Nesta seção são apresentados os resultados obtidos com o algoritmo PSO. Para testar a metodologia proposta, foram utilizados três 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", bem ajustados, e 0.4 pontos "ruins". Este critério foi adotado nos trabalhos de Atkinson [1], Biazi [3], Critchely et al.[11] e Peña [16], para conjuntos com até 50 dados.

Os conjuntos de dados com 40, 200 e 500 pontos 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 $\mu_y = 2$. Pontos influentes em conjuntos de dados simulados segundo este critério são difíceis de serem identificados por apresentarem forte mascaramento.

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 resultados encontrados, objetivando a maximização da função objetivo. Cada linha corresponde aos resultados obtidos para cada conjunto de dados. A coluna número de execuções corresponde ao número de execuções realizadas para cada instância. A coluna melhor solução corresponde ao maior valor encontrado para a função objetivo. Na coluna melhor tempo estão registrados os menores tempos de execução, em segundos, relativos à solução final do problema. A coluna média das soluções corresponde a média aritmética de todas as soluções encontradas nas execuções do programa. A coluna melhor iteração corresponde ao menor número de iterações, dentre todas as execuções, necessário para encontrar a solução do problema.

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

Tabela 1: Resultados computacionais

Nº de dados	Nº de execuções	Melhor solução	Média das soluções	Melhor tempo(s)	Tempo médio(s)	Melhor iteração
40	500	$1,217 \times 10^4$	$1,202 \times 10^4$	2,75	2,84	162
200	500	$3,417 \times 10^4$	$3,186 \times 10^4$	51,19	58,48	701
500	50	$8,162 \times 10^4$	$8,004 \times 10^4$	3024,2	3325,5	1751

Tabela 2: Valores das perturbações (ϵ_i)

$n = 40$		$n = 200$		$n = 500$	
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**

Os resultados apresentados na tabela 2 indicam os pesos (perturbações) associados a cada ponto quando a função objetivo $Fo = |g_{Cook}(\epsilon) - g_{Cook}(0)|$ 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

Neste trabalho, foi aplicada a metaheurística Otimização por Enxame de Partículas na identificação de pontos influentes em modelos de regressão. A metodologia foi aplicada em conjuntos de dados simulados. Os resultados obtidos mostram que a função de sensibilidade de casos é eficiente para detectar pontos influentes. A vantagem do uso desta metodologia está na possibilidade de tratar grandes conjuntos de dados, algumas vezes inviável por outras técnicas.

Pode-se concluir que a metodologia proposta identifica corretamente os subconjuntos de dados, mesmo quando há "mascaramento" de pontos.

Abstract. In this work it was applied Particle Swarm Optimization metaheuristic to identify of influential points. These points exert great influence in determining the coefficients of regression model. The multimodal Case Sensitivity function $g_{Cook}(\epsilon)$ was used as target function. The efficiency of the proposed methodology was tested against sets of simulated data. The results show that this methodology gives satisfactory solutions in the search for influential points.

Referências

[1] R.A. Atkinson, Masking unmasked, *Biometrika*, **73** (1986), 533–541.

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

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

[4] N. Billor, R. Loynes, Local influence: a new approach, *Communs Statist.*, **22** (1993), 1595–1661.

[5] R. Cook, Detection of influential observations in linear regression, *Technometrics*, **19** (1977), 15–18.

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

[7] R. Cook, Assessment of local influence, *J.R.Statist.Assoc.*, **48** (1986), 133–169.

[8] A. Costa, E. Biazi, J. Vítor, Aplicação da metaheurística PSO na identificação de pontos influentes por meio da função de sensibilidade de casos,*Anais do CNMAC*, **2** (2009), 586–591.

[9] K.D. Crawford, R.L. Wainwright, Applying genetic algorithms to outlier detection, em "Proceedings of The Sixth International Conference on Genetic Algorithms", Morgan Kaufmann Publishers, 546–550, 1986.

[10] F. Critchley, Discussion of leave-k-out diagnostics for time series by A.G.Bruce and R.D.Martin, *J. R. Statist. Soc.*, **51** (1989) 407–408.

[11] F. Critchley, R.A. Atkinson, G. Lu, E. Biazi, Influence analysis basead on the case sensitivity function, *Royal Statistical Society*, **63** (2001), 307–323.

[12] R. Eberhart, P. Simpson, R. Dobbins, "Computational Intelligence PC Tools", MA Academic Press Professional, Boston, 1996.

[13] R. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, em "Proceedings of Congress on Evolutionary Computation", San Diego, 84–88, 2000.

[14] F. Hampel, The influence curve and its role in robust estimation, *J. Amer. Statist. Assoc.*, **69**, (1974), 383–393.

[15] J. Kennedy, R. Eberhart, Particle swarm optimization, em "Proc. of the IEEE. International Conference on Neural Networks", Piscataway, NJ, 1942–1948, 1995.

[16] D. Peña, A new statistic for influence in linear regression, *Technometrics*, **47**, (2005), 1–12.

[17] R. Poli, Analysis of the publications on the applications of particle swarm optimization, *J.Artif. Evol. Ap* , **1**,(2008), 1–10.

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

[19] Y. Shi, R. Eberhart, A modified particle swarm optimizer, em "Proc. of the IEEE Congress on Evolutionary Computation", Piscataway, NJ, 69–73, 1998.

[20] J. Tolvi, Genetic algorithms for outlier detection and variable selection in linear regression models, *Soft Computing*, **8** (2004), 527–533.

[21] D. Ye, Z. Chen, A new algorithm for high-dimensional outlier detection based on constrained particle swarm intelligence, *Lecture Notes in Computer Science*, **5009**, (2008), 516–523.