http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (24): 8430-8434, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.8430.8434

Dynamic Population Structure based PSO with Granular Computing for Unified Multiple Linear Regression

Chen Su-Fen
Department of Computer Science and Technology, Nanchang Institute of Technology, Nanchang Jiangxi, 330099, China

Abstract: Unified Multiple Linear Regression (UMLR) is a nonlinear programming model that unifies all kind of multiple linear regression models, such as Principal Components Regression, Ridge Regression, Robust Regression and constrained regression. Although, UMLR has exhibited excellent performances in some real applications, the optimization procedure is not satisfying yet. This study proposes a novel Granular Computing-Particle Swarm Optimization (Grc-PSO) algorithm by introducing granular computing into standard PSO which is used for the optimization of the UMLR model. The experimental results show that the solution got by Grc-PSO algorithm is much better to the real situation than other state-of-art algorithms.

Key words: Particle swarm optimization, granular computing, unified multiple linear regression, nonlinear programming

INTRODUCTION

As a well-studied model in the last few decades, Multiple Linear Regression (MLR) has been widely used on mathematical statistics and some related fields. In the multiple linear regression model, the least squares estimate has been used widely. However, there are some shortcomings during the typical usage (Fang *et al*., 1988; Kleinbaum *et al*., 2007). Therefore, many researchers put forward kinds of MLR models such as Principal Components Regression, Ridge Regression, Robust Regression and Constrained Regression. However, the theories of these methods are too complex to be understood by ordinary engineers. In order to solve this problem, Sun *et al*. (2007) proposed an Unified Multiple Linear Regression (UMLR) which unifies all kinds of multiple linear regression into a nonlinear programming model for clear concept and ease of use. As a novel evolutionary computation technology, swarm intelligence is now becoming a new research hotspot. The basic idea is to achieve random optimization algorithm by simulating colony behavior of natural biology. Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) is a new kind of swarm intelligence algorithm behind ant algorithm, genetic algorithm which has already be regarded as an important branch of evolutionary computing. But it also has some faults such as premature, converging too early because of the population structure. So, it is critical that choosing a suitable population structure for PSO. The topic of fuzzy information granulation was first proposed and discussed by Zadeh (1979). There is a fast growing and renewed interest in this topic (Bargiela and Pedrycz, 2002; Inuiguchi *et al*., 2003; Lin, 2003; Lin *et al*., 2002). Granular computing is likely to play an important role in the evolution of fuzzy logic and its applications, but its application of PSO has not been proposed yet. In the evolutionary process of nature, no individual can learn from each other within species because of a number of reasons such as the region restriction. Limited resources only support limited number of orgamisms. All of these provide the chance for the introduction of granular computing.

The study focus on the topological structure of PSO, where a novel neighbor relationship has been defined with granular computing and then a novel PSO algorithm model based granular computing (Grc-PSO) was proposed which is applied on the problem of MLR model proposed by Sun *et al*. (2007). The goal of this study is to improve the performance of MLR model by the proposed Grc-PSO algorithm.

MATERIALS AND METHOD

Constrained nonlinear programming model in multiple linear regression: In multiple linear regression model, the method of Least Squares Estimate is used widely where the best-fitting line for the observed data is calculated by minimizing function $\Sigma^{m}_{i=1}(y_i-\hat{y}_i)^2$. The value of $(y_i-\hat{y}_i)^2$ becomes larger and larger with increasing value of $(y_i-\hat{y}_i)$. This may seriously distort the prediction. Therefore, researchers try to replace function $(y_i-\hat{y}_i)^2$ with $(y_i-\hat{y}_i)$ which makes the change of value more slowly and minimizes the function eventually. This approach is M-estimation, the most common method of robust regression. Assuming that $\rho(y_i-\hat{y}_i) = |(y_i-\hat{y}_i)|$, the regression parameter β is a M-estimation.

A nonlinear programming model of the least squares estimation is obtained by substituting the sum of absolute value for the sum of the squares where:

$$\min f(\beta_0, \beta_1,..., \beta_n) = \sum_{i=1}^{m} | y_i - \beta_0 - \beta_1 x_{21} - ... - \beta_n x_{ni} | \\ a_i \le \beta_i \le b_i \ (i = 1, 2,..., n) \ c \le \sum_{i=1}^{m} \beta_i \le d \quad (1)$$

Granular computing: The topic of fuzzy information granulation was first proposed and discussed by Zadeh (1979). Then granularity was proposed by Hobbs (1985). The concept of granular computing is first proposed by Zadeh (1996) and indicated that information should be granulated into series of granules.

Granular computing has been used in various fields which includes all things such as theories, methodology, techniques, tools with granule in problem solving.

The basic problem in granular computing is the granule construction, presentation, interpretation and application. The basic idea is the use of granule which may be groups, classes or clusters in problem solving. The main reason of research on granular computing is simplification of problem solving which will play a major role in the design of the intelligent systems.

PSO and its modifications: The mathematical description of standard PSO lists as follows. Given a design problem space D, PSO tries to find the global optimal solution by the movement of a swarm of particles. Each particle's position $x = (x_1, x_2,..., x_m)$ is a possible solution, where m is the dimension of D. The movements of many particles, like bird flocking, have great search power.

The particle updates its position x by the velocity $v = (v_1, v_2,..., v_m)$ in a simple way:

$$x = x+v \quad (2)$$

The movement path of the particle x is decided by the variance of velocity v which is also calculated in a self-updating way. The value of velocity is revised by its particle's current optimal position $p = (p_1, p_2,..., p_m)$ and the global optimal position of all particles $g = (g_1, g_2,..., g_m)$. The velocity v is updated as:

$$v_j = \omega v_j+c_1 rand()(p_j-x_j)+c_2 rand()(g_j-x_j) \quad (3)$$

where, v_j is the j-th element of v, rand() is uniform random numbers between 0 and 1, c_1 and c_2 are two positive acceleration constants, usually $c_1 = c_2 = 2$ and ω is a interia weight controlling the influence of a particles previous velocity and resulting in memory effect. In different problems, PSO can also define some extra constants, such as vmax, vmin, xmax, xmin, to restrain the position and velocity range of particles in the search space.

From sociological angel, the Eq. 3 consists of three parts. The first is "inertia" caused by the particle's previous velocity; the second is "cognition", the thinking of particles, referring to the influence of the information about particle itself on its behavior in next iteration; the third is "society" where sharing information and collaboration among particles influence the behavior in the next iteration. Assuming that c_2 equals to zero, the movement of particles only depends on its cognition, furthermore it has strong ability of local optimization. This kind of PSO is called local best PSO (l-best PSO); when c_1 equals to zero, the movement of particles only depends on its society. It processes strong ability of global optimization in contrast with l-best PSO. Researchers call it global best PSO (g-best).

Equation 3 exhibits an unwanted property, when $x_j = p_j = g_j$, its velocity property also approaches zero under the circumstance where the inertia weight ω is less than one. It implies that eventually all particles will stop moving. This behavior does not guarantee convergence to a global best solution. Therefore, Van den Bergh and Engelbrecht (2002) introduced a new algorithm, called the GCPSO, to pro-actively counter this behavior in a particle swarm and to ensure convergence. Let τ be the index of the global best particle. The idea of GCPSO is then to update the position and velocity of particle τ as

$$v_j = -x_j+g_j+\omega v_j+\varphi (1-2rand()) \quad (4)$$

$$x_j = g_j+\omega v_j+\varphi (1-2rand()) \quad (5)$$

where, v_j denotes the velocity of particle τ; x_j is the position of particle τ; g_j is the best solution in the swarm; rand() is a random number between 0 and 1. φ is a gradual change function which is used to decide the next search behavior of particle τ.

Introduction of granular computing into PSO: In an environment where a large number of individuals compete for available resources, Individuals cooperate around finite, limited resources, resulting in the lack of competition between such areas. Then the environment is partitioned into subgroups based on their resource requirements. So, a new behavioral pattern emerges. Species are parts of a population competing with one another under the environment. Localization of competition is introduced by simply sharing resources among competitive individuals.

The study defines the neighbor relationship from the simulation of human society and nature. If a particle's fitness shows little change over a number of iterations, a granule is created with the particle and its closest topological neighbor. Different species in various and separated granules don't interact with each other. Thus an improved PSO algorithm based granular computing (Grc-PSO) is proposed which includes several operations as follows.

Definition of neighbor relationship R: Formally speaking, the standard deviation in particle i's fitness, σ_i, is tracked over a number of iterations (ordinarily was set to 3). When $\sigma_I<\delta$ (the value of δ is decided by the programmer according to its application), the neighbor relationship R is defined as follows:

$$\forall x, y \in U, x\, Ry \text{ if } y = \arg\min_j \{||x_i - x_j||\} \tag{6}$$

$$1<i, j<N, i \neq j$$

Where:

- x_i = The position of particle i
- x_i = The position of particle j
- y = The closest neighbor of particle i
- N = the total number of particles in the swarm

Granules are obtained by granulation of swarm based on the relationship R. Different species in various and separated granules do not interact with each other which can maintain diversity of swarm in the process of iteration.

Radius of granule: R_j signifies the radius of granule s_j and is defined as

$$Rj = \max \{||g_{8j}.x8j||\} \tag{7}$$

Where:

- g_{8j} = The best particle in granule sj
- x_{8j} = Any particle in granule sj, i ≠ g

Granule absorbing particles: Particles are absorbed into a granule when they move into the granule. That is, a particle i will be absorbed into a granule s_j when and sj_2 intersect, when:

$$||g_{8j1}-g_{8j2}||<(R_{j1}+R_{j2}) \tag{8}$$

Where:

- g_{8j1} = The best particle in granule s_{j1}
- g_{8j2} = The best particle in granule s_{j2}
- R_{j1} = the radius of granule s_{j1}
- R_{j2} = the radius of granule s_{8j2}

The Description of the Grc-PSO Algorithm: In summary, the detail steps of Grc-PSO are presented as follows:

Step 1: Initialize the main particle swarm, including the size of main swarm N, the inertia weight ϵ, acceleration constants c_1, c_2, the maximum number of iterations k, the threshold of creating new granules δ

Step 2: Train the main swarm particles using one iteration of the l-best PSO

Step 3: Update the fitness of each main swarm particle

Step 4: Search the main swarm for any particle that meets the partitioning criteria. If any is found, granules are obtained by granulation of swarm based on the relationship R

Step 5: For each granule:

- Train particles in the granule using one iteration of the GCPSO algorithm
- Update each particle's fitness
- Update the radius of granule

Step 6: If possible, merge granules

Step 7: Allow granules to absorb any particles from the main swarm that moved into it

Step 8: Repeat step 2 until stopping criteria are met.

RESULTS AND DISCUSSION

There are 10 simulations have been done with the Grc-PSO algorithm in the following two examples, an economic dataset and a simulation function. In the examples, the inertia weight ω was scaled linearly from 1.0 to a small value over a maximum of 1,000 iterations of the Grc-PSO algorithm. This behavior results in a rapid initial exploration of the search space that gradually becomes more and more focused. N denotes the initial number of particles in the main swarm before any granule arises and is initialized to 50; δ is the granule creation threshold which is initialized to 0.0001. Other parameters presented in Table 1.

Analysis on economic dataset: Analyses of French economic dataset. This example considers "total import" value as the dependent variable y and "gross domestic

Table 1: Parameters of the Grc-PSO algorithm

Examples	C1	C2	$[x_{min}, x_{max}]$	$[-v_{max}, v_{max}]$
Economic dataset	1.4	1.8	[-100,100]	[-0.1,0.1]
			[0,1.0]	[-0.1,0.1]
Simulation function	1.4	1.8	[-2,2]	
			[-4,4]	[-7,7]

Table 2: Economic dataset used in experiments

Year	x_1	x_2	x_3	y
1974	149.3	4.2	108.1	15.9
1975	161.2	4.1	114.8	16.4
1976	171.5	3.1	123.2	19.0
1977	175.5	3.1	126.9	19.1
1978	180.8	1.1	132.1	18.8
...	...	...	...	...

product" value, "storage" and "total consumption" value as three independent variables x_1, x_2, x_3 with the measure of one billion franc. The data were gathered during 11 years, from 1974-1985, where some data are presented in Table 2.

Kleinbaum *et al.* (2007) lists all possible parameters of regression model. In three-element linear regression equation, it is unreasonable that some parameters are negative. Therefore, researchers remove unreasonable parameters by Principal Components Regression and Ridge Regression with result as follow:

$$\beta_1 = 0.0653,\ \beta_2 = 0.5859,\ \beta_8 = 0.156$$

$$\beta_1 = 0.0727,\ \beta_2 = 0.6091,\ \beta_8 = 0.1062 \quad (9)$$

Considering the physical meaning of parameter b_i, it should be larger than zero. The nonlinear programming model for example 1 is produced as follow:

$$\min f(\beta_0, \beta_1, \ldots, \beta_n) = \sum_{i=1}^{m} |y_i - \beta_0 - \beta_1 x_{21} - \ldots - \beta_n x_{ni}|$$

$$0 \le \beta_i \le b_i (i = 1, 2, \ldots, n) \le \sum\nolimits_{i}^{m} \quad (10)$$

This study does simulations with Grc-PSO algorithm for equation 10. Eventually several solutions are achieved. From these solutions, the best one was chosen.

$$\beta_1 = 0.023878,\ \beta_2 = 0.660411,\ \beta_3 = 0.178346$$

The result above shows that the two sets of results are very close and there is no negative value.

Analysis on simulation data: To make the comparative results sounder, a simulation data are also used in our study. Supposing y is specified by the following function:

$$y = 10+2x_1+3x_2+\varepsilon \quad (11)$$

Table 3: Observed data produced by the Grc-PSO algorithm

β_0	10	10	10	10	10	10	10	10	10	10
x_1	1.1	1.4	1.7	1.7	1.8	1.8	1.9	2.0	2.3	2.4
x_2	1.1	1.5	1.8	1.7	1.9	1.8	1.8	2.1	2.4	2.5

This study has made ten experiments with some observed data presented in Table 3. The normal random residual achieved by simulation lists below:

$$\varepsilon = (0.8\text{-}0.4,\ 0.4\text{-}0.5,\ 0.2,\ 1.9,\ 1.9,\ 0.6,\ -1.5,\ -0.5)$$

According to the Eq. 11, some computed values of the function y listed as follows:

$$y = (16.3, 16.8, 19.2, 18.0, 19.5, 20.9, 21.1, 20.9, 20.3, 22.3)$$

The regression equation which was achieved by multiple linear regression in Fang *et al.* (1988) lists as follows:

$$y = 112.92+11.307x_1+6.591x_2$$

The equation above deviates from original equation largely for x'x is close to strangeness. When the method of Ridge Regression was applied, the parameters of equation varies greatly, where $\beta_{i(k)}$ tends to zero with the gradual increase of k.

The Sun *et al.* (2007) achieves all parameters of Eq. 1 by complex optimal method listed as follows:

$$\beta_1 = 3.428,\ \beta_2 = 1$$

This result shows better results than other regression methods, but there is also a little deviation from original equation. Therefore, this study made an experiment on this example by Grc-PSO algorithm with the best solution of parameters listed as follows:

$$\beta_1 = 1.424,\ \beta_2 = 3.0036$$

This result is close to original model and displays that the Grc-PSO algorithm is efficient.

Compared to the experimental results reported by Sun *et al.* (2007), the observed data produced by the Grc-PSO algorithm are much better. Both two experimental results show that the solution got by the Grc-PSO algorithm is more close to the real situation. Therefore, it is suggested that Grc-PSO has excellent performance on this kind of problem.

CONCLUSION

The constrained linear programming problems are very complex, but there exist many mature algorithms on applications due to its simple concept. This study resolves the constrained linear programming problems which diverse from multiple linear regression by the Grc-PSO algorithm. The experimental results show that the proposed Grc-PSO algorithm has better optimization performance than other state-of-art algorithms. This study combines the Grc-PSO algorithm with the multiple linear regression and achieves a remedy to multiple linear regression.

The future study is to resolve other regression problems such as the non-stationary seepage problem of reservoir.

ACKNOWLEDGMENT

This study was supported by the Natural Science Foundation of China under Grant No. 61105053, the Scientific Research Fund of Jiangxi Provincial Education Departments under Grant No. GJJ12644 and the Nature Science Foundation of Jiangxi Province under Grant No. 20132BAB201043.

REFERENCES

Bargiela, A. and W. Pedrycz, 2002. Granular Computing: An Introduction. Kluwer Academic Publishers, Boston, USA.

Fang, K.T., H. Quan and Q.Y. Chen, 1988. Practical Regression Analysis. Science Press, Beijing, China.

Hobbs, J.R., 1985. Granularity. Proceedings of the 9th February 12, 2014International Joint Conference on Artificial Intelligence, August 18-23, 1985, Los Angeles, USA., pp: 432-435.

Inuiguchi, M., S. Hirano and S. Tsumoto, 2003. Rough Set Theory and Granular Computing. Springer, Berlin, Germany.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proceedings of International Conference on Neural Networks, Volume 4, November 27-December 1, 1995, Perth, Western Australia, pp: 1942-1948.

Kleinbaum, D.G., L.L. Kupper, A. Nizam and K.E. Mulle, 2007. Applied Regression Analysis and Multivariable Methods. 4th Edn., Cengage Learning, USA., ISBN-13: 9780495385257, Pages: 160.

Lin, T.Y., 2003. Granular Computing, Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. Springer, Berlin, Germany, pp: 16-24.

Lin, T.Y., Y.Y. Yao and L.A. Zadeh, 2002. Data Mining, Rough Sets and Granular Computing. Physica-Verlag, Heidelberg, Germany.

Sun, H., Z. Zhang and H. Ge, 2007. Application of PSO to improve multiple linear regression. Comput. Eng. Appl., 43: 43-44.

Van den Bergh, F. and A.P. Engelbrecht, 2002. A new locally convergent particle swarm optimiser. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Volume 3, October 6-9, 2002, Hammamet, Tunisia, pp: 6-6.

Zadeh, L.A., 1979. Fuzzy Sets and Information Granularity. In: Advances in Fuzzy Set Theory and Applications, Gupta, N., R. Ragede and R. Yager (Eds.). North Holland, Amsterdam, pp: 3-18.

Zadeh, L.A., 1996. The key roles of information granulation and fuzzy logic in human reasoning. Proceedings of the IEEE International Conference on Fuzzy Systems, September 8-11, 1996, New Orleans, Louisiana.