Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 277–283
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6247

A nonlinear system modeling method based on projection pursuit and particle swarm optimization algorithm

Zahraa Tariq Mohammed Taher

Ninevah University, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

Based on projection pursuit and particle swarm algorithm, a new method of establishing a nonlinear system model and its algorithm implementation are proposed, and two simulation examples are given. The simulation results show that the method proposed in this paper is used to establish a nonlinear system model. It has the advantages of high prediction accuracy and good engineering practicability.

Keywords: projection pursuit, particle swarm optimization algorithm, nonlinear system, modeling
2020 MSC: 93C10, 68W50

1 Introduction

Traditional statistical models all adopt a confirmatory data analysis idea of "Assumption 2 Simulation 2 Prediction". It is difficult to adapt to the ever-changing objective world, and it is impossible to truly find the inner law of the data. When it is used for high-dimensional, non-linear, non-linear, It is difficult to get good results when predicting and modeling normally distributed data.

In response to this shortcoming, since the 1970s, with the development of computer technology, the international statistical community has developed a projection pursuit technology (Projection Pursuit, referred to as PP) [3].

It adopts a new idea of "Exploratory Data Analysis" such as "Examining Data 2 Simulating 2 Prediction". Its essence is to find the characteristic projection direction from high-dimensional data to low-dimensional data. Through several projection directions Understand the distribution and structure of high-dimensional data. Projection pursuit is an emerging statistical method suitable for the analysis and processing of high-dimensional, nonlinear, and non-normal problems. Therefore, it has always attracted domestic and foreign statistical experts and signal processing scholars. The attention of has been successfully applied in many fields [5, 6].

In 1981, Friedman and Stuetzle proposed a multiple smoothing regression technique to realize Projection Pursuit Regression (PPR) [2], but this method is complicated to calculate, difficult to program, and difficult to promote. In order to better find the projection Based on projection pursuit and genetic algorithm, this paper proposes a new method for establishing regression prediction model of nonlinear system. Two simulation examples are used to verify the effectiveness of the new method.

Email address: zahraa.mohammed@uoninevah.edu.iq (Zahraa Tariq Mohammed Taher)

Received: December 2021 *Accepted:* February 2022

2 Principle of Projection Pursuit Regression

Projection pursuit is a new method for analyzing and processing non-normal and non-linear data. Its basic idea is to use computer technology to project high-dimensional data onto a low-dimensional subspace through a certain combination, and pass the polarities. Minimize a certain projection index and find a projection that can reflect the structure or characteristics of the original data to achieve the purpose of research and analysis of high-dimensional data. The projection pursuit regression model is as follows:

Let $y = f(X)$ *and* $X = (x_1, x_2, \ldots, x_p)$ be one-dimensional and p-dimensional random variables, respectively. In order to objectively reflect the characteristics of high-dimensional nonlinear data structure, projection pursuit regression adopts a series of ridge functions. To approximate the regression function, that is:

$$f(X) \approx \sum_{m}^{M} G_m(Z_m) = \sum_{m=1}^{M} G_m\left(a_m^T X\right) = \sum_{m}^{M} G_m\left(\sum_{j=1}^{p} a_{\mathrm{mj}} x_j\right) \tag{2.1}$$

Among them, $G_m(Z_m)$ is the *m-th* ridge function, M is the number of ridge functions; $Z_m = a_m^T X$ is the independent variable of the ridge function, which is the projection of the p-dimensional random variable X in the direction of am, and am is the projection Direction. The projection pursuit regression model still uses the least squares method as the minimization criterion, that is, select the appropriate combination of the parameters a_{mj}, Gm and the number of ridge functions M in formula (2.1), so that the following formula:

$$L_2 = E\left[y - \sum_{m}^{M} G_m\left(\sum_{j=1}^{p} a_{\mathrm{mj}} x_j\right)\right]^2 \tag{2.2}$$

Reach the minimum.

At present, the projection pursuit regression model generally adopts the multiple smooth regression technology proposed by Friedman and Stuetzle [6]. In view of the fact that this method involves a lot of complex mathematical knowledge and is not easy to program, it limits its application in practical engineering. To this end, this article intends to Based on projection pursuit and genetic algorithm, a new nonlinear system modeling method is proposed.

3 Non-linear system modeling method based on projection pursuit and genetic algorithm

Suppose the mathematical model of the nonlinear system is:

$$f(X) = \sum_{m=1}^{M} \sum_{i=0}^{r} C_{\mathrm{mi}} h_{\mathrm{mi}}\left(a_m^T X\right) \tag{3.1}$$

Among them, r is the order of the polynomial, C is the polynomial coefficient, and h_m represents the orthogonal Chebyshev polynomial, which is calculated in a recursive form:

$$\begin{aligned} &h_{m0}(z) = 1, \quad h_{m1}(z) = z \\ &h_{mr}(z) = 2z h_{m,r-1}(z) - h_{m,r-2}(z), r = 2, 3, \ldots \end{aligned} \tag{3.2}$$

Estimate the parameters in equation (3.1) according to the sample value, determine the regression function $f(X)$, and perform regression prediction. For the nonlinear system model in equation (3.1), the steps to achieve projection pursuit regression in this article are as follows:

Step 1: Determine the number of ridge functions M.

Step 2: Select M mutually orthogonal projection directions $a_1,\ a_2,\ \ldots,\ a_M$, and establish a preliminary regression model

$$f(X) = \sum_{m=1}^{M} \sum_{i=1}^{r} C_{\mathrm{mi}} h_{\mathrm{mi}}\left(a_m^T X\right) \tag{3.3}$$

Step 3: Group optimization.

That is, a_{mj} $(j = 1, 2, \ldots, p)$ and G_m (i.e. h_{mi} $(i = 0, 1, \ldots, r)$) are grouped into one group, $m = 1, 2, \ldots, M$, there are M groups. Remove In addition to one group, use the value obtained in step 2 as the initial value for the other M-1 group, and optimize the remaining group of parameters. After obtaining the result, use the extreme point of this group of parameters as the initial value, Select another set of parameters to optimize, repeat several times until the last selected set of parameter values, so that the formula (2.2) no longer decreases. Each optimization uses the *Matlab* optimization toolbox to solve the nonlinear least squares function.

Step 4: Parameter processing and output regression model:

$$f(X) = \sum_{m=1}^{M} \sum_{i=0}^{r} C_{\mathrm{mi}} H_{\mathrm{mi}} \left(\widehat{a}_m^T X\right) \tag{3.4}$$

In step 3, the projection direction obtained by the Least sq function does not meet the constraint condition $\sum_{j=1}^{p} (a_{\mathrm{mj}})^2 = 1$, for this reason, the ridge function G_m can be processed to convert the orthogonal Chebyshev polynomial h_m into the same order The algebraic polynomial H_{m} of, and make the projection direction meet the constraint conditions. Here, the expression of H_{mi} is:

$$H_{mi}(z) = z^i \qquad i = 0, 1, \ldots, r \tag{3.5}$$

Among the above 4 steps, steps 1, 3, and 4 are not difficult to achieve. The following focuses on the implementation process of step 2.

3.1 Establishment of Stepwise Regression Model

When establishing a preliminary regression model, constructing a suitable algorithm framework is the key to solving the problem. In this paper, the p-dimensional data is projected into the orthogonal projection direction, and the genetic algorithm is used to establish the preliminary regression model.

a) **Construct the orthographic projection direction**

The establishment of the preliminary regression model is based on M standard orthogonal projection directions. For this reason, it is necessary to solve the generation method of standard orthogonal projection directions. For any given p-1 dimensional vector $\theta = [\theta_1, \theta_2, \ldots, \theta_{p-1}]$ T, can generate orthonormal projection directions $a_1, a_2, \ldots, a_p$ in p-dimensional space [7]:

When $j = 1$, there are

$$a_1 = \begin{bmatrix} \cos\theta_1 \\ \sin\theta_1 \cos\theta_2 \\ \sin\theta_1 \sin\theta_2 \cos\theta_3 \\ \cdots \\ \left(\prod_{i=1}^{q-1} \sin\theta_i\right) \cos\theta_q \\ \cdots \\ \left(\prod_{i=1}^{p-2} \sin\theta_i\right) \cos\theta_{p-1} \\ \prod_{i=1}^{p-2} \sin\theta_i \Big) \sin\theta_{p-1} \end{bmatrix} \tag{3.6}$$

When $j = 2, 3, \ldots, p-1$, there are

$$a_j = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ -\sin\theta_{j-1} \\ \cos\theta_{j-1} \cos\theta_j \\ \cos\theta_{j-1} \sin\theta_j \cos\theta_{j+1} \\ \cdots \\ \cos\theta_{j-1} \left(\prod_{i=1}^{q-1} \sin\theta_i\right) \cos\theta_q \\ \cdots \\ \cos\theta_{j-1} \left(\prod_{i=j}^{p-2} \sin\theta_j\right) \cos\theta_{p-1} \\ \cos\theta_{j-1} \left(\prod_{i=1}^{p-2} \sin\theta_i\right) \sin\theta_{p-1} \end{bmatrix} \tag{3.7}$$

When $j = p$, there are

$$a_p = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ -\sin\theta_{p-1} \\ \cos\theta_{p-1} \end{bmatrix} \tag{3.8}$$

It is not difficult to prove that a_j ($j = 1, 2, \ldots, p$) is a pair of orthogonal unit vectors, therefore, they constitute a set of normal orthogonal projection directions in the p-dimensional space.

b) **Use Particle Swarm optimization algorithm to build a preliminary regression model**

Choose M orthogonal projection directions to fit the ridge function. In this paper, a real-coded Particle Swarm optimization algorithm is used to establish a preliminary regression model. The implementation steps are as follows:

Step 1: Determine the population size N ($N = 8\ k + 1, k \in Z^+$), mutation probability, crossover probability, and maximum allowable evolutionary algebra T;

Step 2: Randomly generate N p-1 dimensional θ vectors as the initial population (limit the value range of each component of θ to $-\pi_2,\ \pi_2$);

Step 3: Calculate the fitness value of each individual in the group, the method is as follows:

a. Generate p orthogonal projection directions $a_{0j}, j = 1, \ldots, p$ according to formula (3.6) to (3.8) for the individual θ;
b. Calculate the one-dimensional projection value $Z_{0,j} = a_{0j}^T X (j = 1, \ldots, p)$
c. From the p orthogonal projection directions, select M projection directions to make the regression model

$$f(X) = \sum_{m}^{M} G_m\left(a_m^T X\right) = \sum_{m=1}^{M} \sum_{i=0}^{r} C_{\mathrm{mi}} h_{\mathrm{mi}}\left(a_m^T X\right)$$

The fitting residual of is the smallest. Among them, a_m ($m = 1, 2, \ldots, M$) are the M projection directions selected from a_{0j}(j = 1, ..., p), and the ridge function G_m uses the orthogonal contract Bishev polynomial fitting;

d. Calculate the fitted value $\widehat{y}$ according to the regression model;
e. Calculate the objective function: $Q = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \widehat{y}_i\right)^2$
f. Define the fitness value function of the individual as

$$f(Q) = \frac{1}{Q \mp 0 \cdot 001}$$

And calculate the fitness value of each individual in the group.

Step 4: Save the best individual P_{best} and G_{best} ; and use the following equations :

$$\begin{gathered} V = \&w^{\star}V + c_1^{*} \text{ rand }^{\star}(P_{\text{best}} - X_{t-1}) \\ \& + c_2^{*} \text{ rand }^{\star}(G_{\text{best}} - X_{t-1}) \qquad (11) \\ \& X_t = X_{t-1} + V \end{gathered} \tag{3.9}$$

Step 5: According to the fitness value of the individual, eliminate $\frac{N-1}{4}$ worst individuals in the group, and randomly rearrange the remaining $\frac{3(N-1)}{4}$ individuals in the group;

Step 6: Perform a crossover operation on $\frac{3(N-1)}{4}$ individuals randomly rearranged, and then perform a mutation operation according to the mutation probability;

Step 7: Perform $\frac{N-1}{4}$ on the optimal individual PB according to the mutation strategy Sub-mutation, resulting in N-14 new individuals;

Step 8: Combine the optimal individual PB, the $\frac{3(N-1)}{4}$ individuals produced in **step 6** and the $\frac{N-1}{4}$ individuals produced in **step 7** together to form a new generation group;

Step 9: Check whether the current population meets the algorithm termination condition? If not, go to **step 3**; if it is satisfied, output the regression model and end the algorithm.

4 Experimental Results

According to the above algorithm steps, this article uses *Matlab* language to compile the corresponding modeling program, and test the performance of the algorithm. In order to verify that the algorithm proposed in this article can better solve the modeling problem of nonlinear system, the following two Experiment .

Experiment 1.

Establish a river water pollution prediction model for a certain section of the Luo River, and the prediction index y is the BOD concentration. There are 7 factors related to y, namely $x_1,\ x_2,\ \ldots,\ x_7$. Table 1 shows 15 sets of monitoring data, which are quoted from literature [6].

It can be seen from Table 1 that there is a high-dimensional nonlinearity between the factor monitoring data and the BOD monitoring data y. To this end, the algorithm proposed in this paper is used to establish a 7-factor regression prediction model, and the first 12 samples are used to estimate the projection in the projection pursuit regression equation for the direction and polynomial coefficients, the latter 3 samples are reserved for prediction testing. The number of ridge functions is 5, the order of the polynomial is 3, and the regression prediction equation established by the method proposed in this article is:

$$y = \sum_{m=1}^{5} \sum_{i=0}^{3} C_{\mathrm{mi}} H_{\mathrm{mi}} \left(\sum_{j=1}^{7} \widehat{a}_{\mathrm{mj}} x_j \right) \tag{4.1}$$

in ,

$$\{\mathcal{C}_{mi\}}\}_{5\times 4} = \begin{bmatrix} 522.2351 & -1082.4769 & 691.0824 & -139.5877 \\ 11.1254 & 105.6509 & 293.4419 & 173.5632 \\ 0.9010 & 0.7836 & -0.8189 & -0.65765 \\ -1.9148 & 49.5533 & -363.0012 & 797.8190 \end{bmatrix}$$

$$\{\widehat{a}_{\mathrm{mj}}\}_{5\times 7} = \begin{bmatrix} 0.0126 & 0.0356 & 0.0159 & 0.0818 & 0.6165 & 0.0157 & 0.6618 \\ 0.0383 & -0.0301 & -0.0172 & -0.0219 & 0.2917 & -0.0812 & -0.9304 \\ -0.0678 & 0.5422 & -0.0223 & 0.1209 & -0.32214 & -0.0312 & 0.7197 \\ -0.02234 & -0.0392 & 0.0013 & -0.0150 & -0.1097 & -0.0139 & 0.8833 \\ 0.0042 & 0.0321 & 0.0114 & -0.0139 & -0.0295 & -0.0215 & 0.8845 \end{bmatrix}$$

Table 1: Measured values of pollution concentration and related factors of Tigris River

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
1	6.96154	-0.17614	27.01209	6.764651	1.140846	4.843784	0.116294	9.448842
2	6.167901	-2.1514	27.58627	4.798907	1.176498	1.93634	0.129707	12.354
3	2.238891	-3.01533	26.04843	4.784265	1.184031	4.126044	0.14782	15.67069
4	5.020052	-0.66336	26.08449	8.62352	1.161703	3.723441	0.075523	5.979949
5	7.976544	-2.25165	26.02094	8.588187	1.140598	3.72844	0.153221	6.368785
6	2.441257	-1.5874	15.05523	1.54386	1.094793	4.365894	0.159903	4.041452
7	1.958995	-1.28391	15.86299	1.559516	1.568207	4.358556	0.18941	3.806484
8	1.072768	-2.04702	17.1032	1.539912	1.390571	4.331338	0.138788	4.056396
9	1.267952	-1.83092	17.56147	1.54358	1.394204	4.29776	0.148603	4.06182
10	0.980135	-0.17177	17.03624	3.674518	1.006646	2.059859	0.109424	2.790022
11	1.602784	-0.0131	17.00495	3.642393	1.052096	2.033393	0.121711	1.847812
12	2.829809	-1.11186	13.54896	3.319036	1.047371	1.143089	0.170281	2.595963
13	2.190085	-1.20717	13.51925	3.3553	0.995208	1.233914	0.137083	2.72567
14	2.377466	-0.54199	14.51231	3.737393	0.95312	0.600131	0.193849	1.692189
15	2.044518	-0.5983	14.52055	3.677029	0.932866	0.599553	0.115816	2.393585

In this simulation example, the fit test of the first 12 samples and the prediction test effect of the last three reserved samples are shown in the 5th and 6th columns in Table 2. For the convenience of comparison, the 3rd and 4th columns in Table 2 The column lists the fitting and prediction test effects of the projection pursuit regression model (PPR model) given in the literature [6]. The results listed in Table 2 show that the fitting effect of the model built in this paper is better than that of the PPR in the literature [6] The fitting effect of the model is slightly worse, but the prediction effect is better than the PPR model in [6]. Generally speaking, the effects of the two models are relatively ideal.

Experiment 2

The problem of positioning controller modeling encountered by the author in scientific research can be attributed to the establishment of a regression prediction model for a high-dimensional nonlinear system. In order to establish a positioning controller model, the modeling data set X $\in R^{250\times13}$ is now given, y $\in R^{250\times1}$, check the data set $\widehat{\mathrm{X}} \in R^{500\times13}$,$\widehat{y} \in R^{500\times1}$, verify the positioning accuracy of the positioning controller after modeling with the algorithm in this paper. Take the ridge function The number is 4, and the order of the polynomial is 3. The obtained 4 projection pursuit directions are shown in Table 3. The established regression prediction equation is:

$$y = \sum_{m=1}^{4}\sum_{i=0}^{3} C_{\mathrm{mi}} H_{\mathrm{mi}} \left(\sum_{j=1}^{13} \widehat{a}_{\mathrm{mj}} x_j\right)$$

In,

$$\{\mathcal{C}_{mi\}}\}_{4\times4} = \begin{bmatrix} -0.1666 & 11.7745 & -191.0544 & 578.2155 \\ 35.2819 & -171.0736 & 313.2918 & -179.8302 \\ -0.4015 & -32.6591 & -755.8110 & -5052.8766 \\ 4.1917 & -151.5139 & 2033.2851 & -6826.9186 \end{bmatrix}$$

Table 2: Comparison of the effect of fitting and prediction test

Sample number	y	The PPR model built in [7]		Paper model	
		y_{ppr}	Relative error(%)	$\widehat{y}$	Relative error(%)
1	9.367567	9.441494	0.006205	9.398631	-0.24624
2	12.32089	12.29715	-0.07018	12.33203	0.310787
3	15.69052	15.63625	0.104635	15.59684	0.024125
4	5.947539	5.891375	0.150543	6.031792	1.460481
5	6.386847	6.408172	0.476143	6.405347	0.092789
6	4.091213	3.973328	-0.63689	4.08661	0.914954
7	3.770401	3.900719	1.408989	3.855971	0.658523
8	4.054555	4.015666	-0.29191	4.118725	1.809048
9	4.053627	4.010337	0.577724	3.897138	-3.9701
10	2.836186	2.726737	-2.90949	2.731798	-2.69657
11	1.848419	1.942606	0.953377	1.971253	3.935133
12	2.619721	2.572167	0.110915	2.637667	0.887289
13	2.749994	2.825539	0.282581	2.753652	-0.22115
14	1.653412	1.963209	17.73381	1.872159	13.83984
15	2.38126	1.981735	-17.2706	2.149018	-12.1357

Table 3: Projection pursuit direction

Projection direction	Forecast factor						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$\widehat{a}_1$	0.00739	0.015133	0.000594	389.9391	3507.004	0.471488	-0.08701
$\widehat{a}_2$	0.00506	0.023796	0.022958	0.025202	0.097956	-0.22578	0.27172
$\widehat{a}_3$	0.002829	-0.00498	2.72E-05	0.084624	-0.00657	0.040165	0.374983
$\widehat{a}_4$	0.000372	-0.0156	0.040764	0.051778	-0.03119	0.066543	0.128062

Projection direction	Forecast factor						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$\widehat{a}_1$	0.628975	-0.08287	-0.47298	0.044378	-0.07422	0.09144	-
$\widehat{a}_2$	0.065464	0.112032	-0.17963	-0.37097	0.741204	0.366241	-
$\widehat{a}_3$	-0.13492	-0.50864	0.101338	0.542762	-0.46922	0.215324	-
$\widehat{a}_4$	0.50231	0.384544	0.131667	-0.31018	-0.01053	0.689628	-

The test results of the regression prediction model of the positioning controller established in Example 2 are as follows:

(1) The fitting residual of the modeling data set is shown in Figure 1, and the residual standard deviation $\sigma_s = 0.029$;

(2) The test residual of the test data set is shown in Figure 2. The residual standard deviation is $\sigma t = 0.0391$. In engineering practice, the requirement for the positioning accuracy of the positioning controller is the residual standard deviation $\sigma < 0.068$, which can be seen in this paper. The method establishes the regression prediction model of the positioning controller to meet the accuracy requirements.

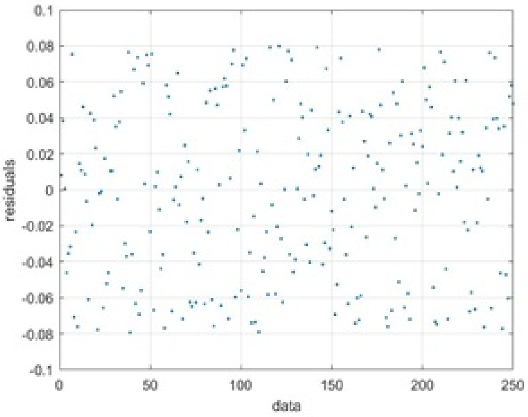

Figure 1: Modeling data set fitting residual

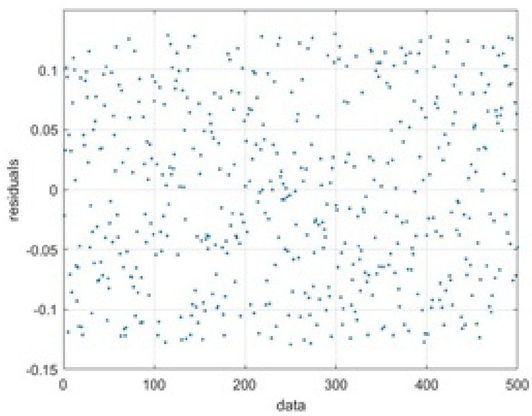

Figure 2: Test residuals of the test data set

5 Conclusion

Based on projection pursuit and particle swarm optimization algorithm, this paper proposes a new algorithm for establishing regression model of nonlinear system. This algorithm has the advantages of simplicity and ease of programming. Simulation examples show that the algorithm in the article can effectively solve the modeling of high-dimensional nonlinear systems. Problems, showing good prospects for engineering applications.

References

[1] A. Berro, S.L. Marie-Sainte and A. Ruiz-Gazen, *Genetic algorithms and particle swarm optimization for exploratory projection pursuit*, Ann. Math. Artif. Intell. **60** (2010), no. 1, 153–178.

[2] J.H. Friedman and W. Stuetzle, *Projection pursuit regression*, J. Amer. Statist. Assoc. **76** (1981), 817–823.

[3] J.H. Friedman and J. W. Tukey, *A projection pursuit algorithm for exploratory data analysis*, IEEE Trans. Comput. **23** (1974), 881–889.

[4] Q. Jiang, Q. Fu and Z. Wang, *Comprehensive evaluation of regional land resources carrying capacity based on projection pursuit model optimized by particle swarm optimization*, Trans. Chinese Soc. Agric. Engin. **27** (2011), no. 11, 319–324.

[5] Z. Li, X. Deng and Y. Xin, *Model for drought and flood trend prediction using projection pursuit regression*, J. Natural Disasters **6** (1997), no. 4, 68–73.

[6] Z. Li, *The PPR model for pollutant concentration prediction*, Environ. Sci. **18** (1997), no. 4, 38–40.

[7] J. Li and Q. Chen, *The construction of a continuously changeable base of n2demensional space and projection pursuit*, J. Guizhou Instit. Technol. **25** (1996), no. 6, 24–28.

[8] F. Marini and B. Walczak, *Particle swarm optimization (PSO)*, Tutor. Chemomet. Intell. Lab. Syst. **149** (2015), 153–165.

[9] A. Ruiz-Gazen, S. L. Marie-Sainte and A. Berro, *Detecting multivariate outliers using projection pursuit with particle swarm optimization*, Proc. COMPSTAT, **2010** (2010), 89–98.