Genetic Algorithms for Simultaneous Equation Models

José J. López
Universidad Miguel Hernández (Elche, Spain)
Domingo Giménez
Universidad de Murcia (Murcia, Spain)

DCAI 2008

Contents

Introduction

- S.E.M. have been used in econometrics for years. Nowadays they are used in medicine, network simulation, and even in the study of the divorce rate.
- Traditionally, Simultaneous Equation Models (SEM) have been developed by people with a wealth of experience in the particular problem represented by the model.
- The objective is to develop an algorithm which, given the endogenous and exogenous variables, finds a satisfactory SEM.
- The space of the possible solutions is very large and exhaustive search methods are not suitable here.
- A combination between genetic and random search is studied.

Simultaneous Equations Models

The scheme of a system with *N* equations, *N* endogenous variables, *K* exogenous variables and *d* sample size is (*structural* form)

$$Y_1 = \beta_{12}Y_2 + \beta_{13}Y_3 + ... + \beta_{1N}Y_N + \gamma_{11}X_1 + ... + \gamma_{1K}X_K + u_1$$
$$Y_2 = \beta_{21}Y_1 + \beta_{23}Y_3 + ... + \beta_{2N}Y_N + \gamma_{21}X_1 + ... + \gamma_{2K}X_K + u_2$$
$$\dots$$
$$Y_N = \beta_{N1}Y_1 + \beta_{N2}Y_2 + ... + \beta_{NN-1}Y_{N-1} + \gamma_{N1}X_1 + ... + \gamma_{NK}X_K + u_N$$

where Y_i, X_j and u_i are *dx1* $i = 1...N,\ j = 1...K$

These equations can be represented in matrix form

$$BY + \Gamma X + u = 0$$

The problem: Find the best SEM given a set of values of variables

- One model is considered better than another if it has a lower criteria parameter.
- AIC is one of the most used methods for comparing models.

$$AIC = d \ln |\hat{\Sigma}_e| + 2\sum_{i=1}^{N}(n_i + k_i - 1) + N(N+1)$$

- d is the sample size, n_i and k_i the number of endogenous and exogenous variables in equation i, and $\hat{\Sigma}_e$ is the covariance matrix of the errors.

Genetic Algorithms for selecting the best SEM

- Each chromosome represents one candidate.
- A chromosome is defined as a matrix with *N* rows and *N+K* columns.
- In each row, an equation is represented using ones and zeros.
- If variable *j* appears in equation *i*, the value for the *(i,j)* position in the chromosome is one, and zero if not.
- The first *N* columns of a chromosome represent the endogenous variables and the other *K* columns represent the exogenous ones.

For example, in a problem with *N=2* endogenous variables (Y_1 and Y_2) and *K=3* predetermined variables (X_1, X_2 and X_3):

$$y_1 = \beta_{1,2} y_2 + \gamma_{1,1} x_1 + \gamma_{1,2} x_2 + u_1$$

$$y_2 = \beta_{2,1} y_1 + \gamma_{2,3} x_3 + u_2$$

$\longrightarrow$

11110

11001

Defining a valid chromosome

- The model has to have at least one equation.
- If the *(i,i)* element is zero, the column *i* will have only zeros.
- Each equation in the model must have at least two variables.
- Rank condition: Equation *i* is identified if it is possible to find a *(N-1) x(N-1)* matrix with full range where the columns are the unknown variables

$$\gamma_{1,1},\ldots,\gamma_{N,K},\beta_{1,2},\ldots,\beta_{N,N-1}$$

that do not appear in the equation.

- The number of comparisons when evaluating a chromosome is :

$$N^2 + N(N+K) + N\frac{(K+N-2)!}{(K-1)!}$$

Evaluating a chromosome

- The algorithm on the right shows the scheme of the fitness function of a chromosome.

- The cost of evaluating a chromosome is :

$$\approx \mathrm{O}(K^2 Nd + K^3 N)$$

1. **BUILD** the system using chromosome c and the set of variables Y and X
2. **SOLVE** the system
3. **COMPUTE** the error between the variables Y and its estimation
4. **COMPUTE** AIC

Comparison between defining and evaluating a chromosome

- The cost of defining a valid chromosome is lower than the cost of the fitness function. But it is not negligible and must be considered.

Size of the problem			Times		
			Valid	Fitness	
N	**K**	**d**	chromosome	function	sp
10	100	500	0,00027	0,09	322,22
50	100	500	0,07	0,79	11,29
50	100	1000	0,07	1,77	25,29
75	100	500	0,43	1,94	4,51
75	100	1000	0,42	3,50	8,33
100	200	500	1,43	7,70	5,38
100	200	1000	1,42	18,21	12,82
150	200	500	7,29	20,31	2,79
150	200	1000	7,28	45,32	6,23

Initialization and EndConditions

- Each chromosome is generated according to the algorithm on the right.
- The population size (called *PopSize*) is stated at the beginning.
- The process is repeated until it reaches a maximum number of iterations, called *MaxIter*, or the best fitness is repeated over a number of successive iterations, called *MaxBest*.
- Both parameters are stated at the beginning of the algorithm.

1. **GENERATE** the $N(N+K)$ elements randomly (with the same probability of zeros and ones)

{C1 AND C2 CONDITIONS}

2. **IF** N or $N-1$ elements $e_{(i,i)}$ are zero with i=1,…,N
3. invert all the elements $e_{(i,i)}$ with i=1,…,N
4. **END IF**

{C3 CONDITION}

5. **FOR** i=1…N
6. **IF** the element $e_{(i,i)}$ is zero
7. make all the elements zero in column i
8. **END IF**
9. **END FOR**

{C4 CONDITION}

10. **FOR** i=1…N
11. **IF** equation i fails the range condition
12. generate randomly this equation (row i) and go to 2
13. **END IF**
14. **END FOR**

Crossover

- Three sorts of crossover are studied:
  - Single Point (SP)
  - Single Point considering equations (SPCE)
  - Inside an Equation (IE)

parents		**SP** e = 10		**SPCE** e = 1		**IE** e = 2, v1 = 2, v2 = 3	
parent1	parent2	child1	child2	child1	child2	child1	child2
11110110	10100100	11110110	10100110	11110110	10100100	11110110	10100100
11110101	01110100	11110100	01110101	01110100	11110101	11110100	01110101
01110110	11110110	11110110	01110110	11110110	01110110	01110110	11110110

problem size			crossover **SP**			crossover **SPCE**			crossover **IE**		
N	***K***	***d***	t	iter	*best fitness*	t	iter	*best fitness*	t	iter	*best fitness*
10	15	50	3,03	48	2683,13	5,11	97	2732,90	0,66	20	2833,41
15	20	50	8,00	62	4548,68	6,73	53	4540,93	1,94	40	4709,50
30	40	100	58,33	50	21937,02	87,54	72	22120,10	9,47	17	22765,68
40	50	100	325,87	111	30956,78	294,19	102	31262,20	64,41	24	32975,04

Mutation

- A small probability of mutation is considered in each iteration.
- A chromosome of the new subset generated in the crossover is chosen randomly, and an equation and a variable are generated randomly. Then, the element is inverted.
- PROBLEM: When a chromosome is mutated and then situated in a different part of the set of solutions, it does not normally have enough quality to survive to create new chromosomes in this area, and perhaps a better solution is close to it.

Random Search

- To avoid this problem, a random search is used in the mutation, following the algorithm on the right.

- A chromosome is good enough when its evaluation is lower than a parameter called *SV*.

1. Generate *e* between 1 and *N* ramdonly.
2. *EndConditions*=**FALSE**
3. **WHILE** *Not EndConditions*
4. Generate *v* between 1 and *N+K* randomly
5. *c1*=Mutate(*c*) {invert the element *(e,v)* of chromosome *c*}
6. **IF** GoodChromosome(c_1) **AND** Evaluation(c_1)<Evaluation(*c*)
7. *c=c1*
8. **END IF**
9. **IF** Evaluation(*c*)<*SV*
10. EndConditions=TRUE
11. **END IF**
12. **END WHILE**

		N=10, K=20		N=20, K=30	
Mode	**NEG**	**AIC**	**time**	**AIC**	**time**
without random search	-	2138.93	5.10	4658.06	15.41
with random search	1	2143.54	9.79	4710.53	49.14
	[N/2]	1491.13	12.62	3072.98	102.23
	[N/4]	-680.61	27.48	811.65	227.35
	N	-3586.46	34.17	-4920.01	449.78

Experimental Results

- Experimental results have been obtained in a system with two nodes Intel Itanium, connected by Gigabit Ethernet, where each node is equipped with four dual-core 1.4 GHz Montecito processors, i.e. 8 processors per node.
- Comparison of the solution found by the genetic algorithm and the optimum, when varying the population size (*PopSize*), *N* and *K*. The sample size is *d*=10, the crossover is inside an equation.
- Execution time (in seconds) and speed-up of the algorithm in shared memory.

size		best fitness		
N	**K**	***PopSize*=100**	***PopSize*=500**	Optimum
2	2	66,44	66,44	66,44
2	3	46,18	46,18	46,18
3	3	-177,03	-214,91	-216,68
3	4	-124,05	-213,16	-216,68
4	4	-99,73	-161,67	-218,58

PopSize	**N**	**K**	**d**	**1th**	**2th**	**sp**	**4th**	**sp**	**8th**	**sp**
100	10	15	50	4,22	2,51	1,68	1,62	2,60	1,04	4,06
100	30	40	100	40,74	26,21	1,55	16,24	2,51	12,31	3,31
100	50	65	150	217,79	152,19	1,43	102,27	2,13	63,81	3,41
100	70	90	200	709,05	417,62	1,70	277,15	2,56	185,88	3,81
500	10	15	50	21,31	11,55	1,85	7,50	2,84	4,70	4,53
500	30	40	100	201,29	115,71	1,74	62,30	3,23	47,47	4,24
500	50	65	150	1065,77	699,20	1,52	368,11	2,90	229,68	4,64
500	70	90	200	3580,94	1927,76	1,86	1076,45	3,33	699,21	5,12

Conclusions and Future works

Conclusions

- An algorithm to obtain a satisfactory Simultaneous Equation Model from a set of variables is studied.
- Genetic and random search are combined to avoid to fall into local minima and to speed up the convergence.
- A shared memory version, which allows us to efficiently use multicore processors in the solution of the problem, has been developed.

Future Works

- Application to real problems.
- Develop a hybrid (message-passing plus shared memory) algorithm.
- Use and comparison other criteria parameters.

References

- Akaike, H., Information theory and an extension of the maximum likelihood principle. In: B.N. Petrov, Csaki F. (Ed.), Proc. 2nd Int. Symp. on Information Theory, Akademiai Kiado, Budapest, 267 281, 1973.
- Bedrick, E.J., Tsai, C.-L. Model selection for multivariate regression in small samples. Biometrics, 50, 226-231, 1994.
- Bozdogan, H., Haughton, D. Informational complexity criteria for regression models. Computational Statistics and Data Analysis, 28, 51-76,1998.
- Fujikoshi, Y., Satoh, K. Modified AIC and Cp in multivariate linear regression. Biometrika, 84 (3), 707-716, 1997.
- Gorobets,A., The Optimal Prediction Simultaneous Equations Selection, Economics Bulletin, 36(3), 1-8, 2005.
- Gujarati, D. 1995. Basic Econometrics, McGraw Hill.
- Mitchell, M. 1998. An Introduction to Genetic Algorithm, MIT Press.
- Shi, P., Tsai, C.-L., . A note on the unification of the Akaike information criterion. J.R. Statist. Soc. B, 60 (3), 551-558, 1998.