Technological Modelling for Graphical Models:

an approach based on genetic algorithms

Alberto Roverato[*], Sandra Paterlini

Abstract. Automatic model search procedures aim at identifying the model that maximises a given fitness function, thereby regarding model selection as an optimisation problem. However, it is unrealistic to believe that the fittest model represents the best solution to the search problem. In fact, even if it is possible to score all of the candidate models, it hardly happens that there is an unequivocal answer to the question of which model best explains data. In this paper we propose an automatic model search procedure for the identification of an optimal set of good models. In a technological approach to model selection the identified models can co-exist, whereas in a scientific modelling approach such models represent a starting point for further context-dependent analysis. Examples of the application of the proposed procedure to real data are given.

Key-words: model search, scientific modelling, technological modelling, genetic algorithm, log-linear graphical model.

---

[*]Corresponding author:Tel.:+39-0522-233217; fax: +39-0522-233206
E-mail address: roverato.alberto@unimore.it (A.Roverato)

1. Introduction

Model search procedures aim at identifying one model, out of a finite set M of candidate models, which best explains the data D. More precisely, following Cowell *et al.* (1999, p. 244), we could distinguish between *technological modelling* and *scientific modelling*. The technological modelling approach regards the model according to its "usefulness", irrespective of whether it corresponds to any underlying reality. As a consequence, several models can co-exist and there is no reason to constrain a search procedure to identify a unique model. On the other hand, in the scientific modelling approach it is assumed that there exists a "true" model from which data are drawn so that model search procedures should be designed to identify such a unique model.

Automatic model search procedures can be regarded as optimisation algorithms: the elements m of M are scored according to a fitness function $f(m, D)$ which gives a measure of the goodness of fit of the model and has to be optimised. Typically, the dimensionality of M is so large that it is time-unfeasible to evaluate every $m \in M$ and only the models in a very small subset of M can be evaluated. As a consequence, the way such a subset is chosen is a critical aspect that plays a fundamental role in the efficiency and effectiveness of the procedure. Recently, a great effort has been devoted to the research of always more efficient and robust search algorithms often supported by the huge progress in the computational sector. Nevertheless, it is important to underline that model selection is not essentially a computing problem but it is strongly dependent from the context of analysis. Indeed, even if it is possible to

score all of the candidate models, it hardly happens that there is an unequivocal answer to the question of which model best explains data; see for instance Whittaker (1990, p. 241), Christensen (1997, p. 215), and Edwards (2000, p. 157). Our standpoint is that, both in the technological and in the scientific modelling approach, it is always useful that an automatic model search procedure identifies a set of good models (that is, a set of parsimonious models consistent with the data) possibly representing a starting point from which further context-dependent analysis can be developed.

Log-linear graphical modelling (see Lauritzen, 1996) constitutes an important field in which it is required to tackle efficiently and effectively the model search problem. In this context, stepwise procedures have been proposed (see Whittaker, 1990 and Edwards, 2000) and implemented, among others, in the statistical packages CoCo (Badsberg, 1995), MIM (Edwards, 2000), GraphFitI (Blauth and Pigeot, 2000). More recently, stochastic search and optimisation algorithms, such as genetic algorithms, which are usually computationally more expensive than stepwise procedures, but have better explorative and convergence properties, have attracted increasing attention; see Fouskakis and Draper (2002) for a review of the application of such techniques in statistics and Bargelt and Kruse (2002, p. 227), Exteberria *et al*. (1997), Larranaga *et al* (1999) and Poli and Roverato (1998) for applications to structural learning in graphical models.

In this work we propose an automatic model search procedure, based on genetic algorithms, that identifies a set of good models satisfying a given optimality criterion. The cardinality of such a set is not specified a priori and we show empirically that, although the selected models can be ranked according to their fitness value, such a

ranking is independent of the ranking based on the distance from the true generating process. In other words, within the selected set the fitness function does not seem to be able to discriminate between models. As a consequence, to choose one model, out of the set of selected models, on the basis of the fitness function, is not justified. Further analysis on real datasets, coming from economical and socio-medical frameworks highlights the usefulness of the proposed search procedure.

The paper is organized as follows. Sections 2 introduces briefly log-linear models and genetic algorithms, Section 3 describes the proposed evolutionary algorithm. The results from the analysis of simulated datasets are reported in Section 4, while the analyses of real dataset in financial and social environments are reported in Section 5.

2. An outline on the technical framework

In this section we briefly introduce graphical models for discrete data and genetic algorithms. We refer to Lauritzen (1996) for a comprehensive account of the graphical model theory and to Mitchell (1996) and Goldberg (1989) for a detailed description of genetic algorithms.

2.1 Log-linear graphical models

Let $V = \{a, b, ..., m\}$ be a finite set of discrete random variables. In undirected graphical models each variable in V is associated with a vertex of an undirected graph $G = (V, E)$ (Darroch *et al.*, 1980). The edge set $E \subseteq V \times V$ gives the conditional independence structure of V; specifically, each missing edge $(i, j) \notin E$, with $i, j \in V$,

denotes the conditional independence of variables i and j given the remaining variables $V \setminus \{i, j\}$. A graph is *undirected* if $(i, j) \in E$ implies $(j, i) \in E$. A graph is said *complete* if for all $i, j \in V$ with $i \neq j$ it holds that $(i, j) \in E$. A subset A of V induces a subgraph $G_A = (A, E_A)$ where $E_A = V \cap A \times A$. The subset $A \subseteq V$ is called complete if G_A is complete. The subset $A \subseteq V$ is a *clique* of G if it is maximally complete, i.e. A is complete, and if $A \subset B$ then B is not complete. For instance, the graph in Figure 1 has vertex set $V = \{a, b, c, d, e\}$, edge set $E=\{(a,b), (b,c), (b,d), (c,d), (c,e), (d,e)\}$ and its cliques are *{a, b}, {b, c, d}* and *{c, d, e}*.
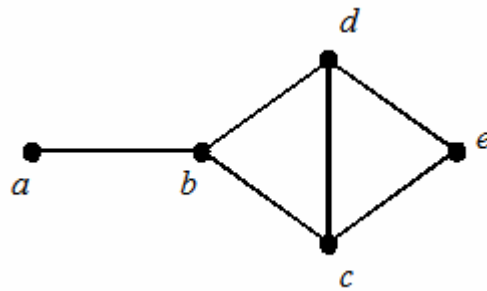
Figure 1: Example of an undirected graph.

An undirected graph is identified by the set of its cliques and, in the case where the random variables in V are all discrete, the model defined by an undirected graph G is the hierarchical log-linear model where the cliques of the graph correspond to the maximal terms in the model.

The set M of all graphical models for $/V/$ variables coincides with the set of all undirected graphs on $/V/$ vertices. Note that the cardinality of M is $2^{\binom{|V|}{2}}$ and it grows exponentially with the number of variables. In a related context, Chickering (1996)

5

showed that the problem of identifying the graphical model with the highest score is NP-hard.

2.2 Genetic Algorithms

Genetic Algorithms (GA) are stochastic search and optimisation algorithms, modelled on the Darwinian principle of the Survival of the Fitness (Holland 1975). They have been widely used in different fields since they allow to deal with complex optimisation problems by exploring effectively the solution domain and converging in a path-independent way towards the optimal solution.

A GA consists of a population of individuals (strings), where each individual represents the mathematical encoding of a candidate solution of the problem under investigation. Each individual is composed by a sequence of cells, which can assume binary or real values. The algorithm starts by randomly generating a population of individuals. Then, each individual is iteratively considered and its goodness is evaluated with respect to a fitness criterion, which drives the evolutionary mechanism. After all the individuals have been processed and the corresponding fitness values are computed, the current population is evolved through operators inspired to biological mechanisms. *Selection*, *crossover*, *mutation* and *elitism* are the evolutionary operators that characterize the structure of the canonical genetic algorithm. Selection allows the transmission of the individuals with higher fitness values in an intermediate population, which then encounter crossover and mutation operators. Crossover consists in exchanging, with a fixed probability p_c, genetic fragments of two mating strings randomly chosen in the intermediate population,

while mutation consists in altering the values of single cells of different individuals with a fixed probability p_m. Different schemes for selection (proportional fitness-based and rank-based are the most common) and crossover (e.g. single point, double-point, uniform) can be chosen for the evolution of the population of candidate solutions (Goldberg 1989, Mitchell 1996). Elitism operator consists in inserting directly in the population of the next generation the best individuals from the current population. Therefore, selection, crossover, mutation and elitism determine the composition of the new population from the current one. The evolutionary mechanism is iteratively repeated until a stopping criterion is fulfilled. A widely used stopping criterion requires that the best individual does not change for a fixed number of generations. Then, the best individual of the last generation corresponds to the optimal solution of the problem under investigation.

3. Model search procedure

In this section we describe the procedure for the identification of an optimal set H of good models. The characterising feature of our selection method is the repeated use of a properly tuned GA. We start with $H=\emptyset?$ and at every run of the GA an adequate model is identified and, if not previously selected, included in H. The optimality criterion satisfied by H derives from the convergence criterion of the overall procedure, hereafter described in details.

The GA evolves populations of candidate solutions to the search problem. Each individual is a binary string determining the structure of an undirected graph G. Every

cell corresponds to one of the $K=\binom{V}{2}$ edges of the complete graph and cells with

unitary value correspond to edges belonging to the graph. The goodness of fit of the

corresponding model for each individual is evaluated in terms of the Akaike's

Information Criterion (AIC, Akaike, 1973, Badsberg 1995, p. 100).

In the implementation of the GA several parameters have to be set a priori. These

determine the rules for the generation of new candidate solutions to the search

problem and an optimal balance between *exploration* (the search for new individuals)

and *exploitation* (the use and propagations of previously identified individuals) must

be found. In other words, the system has to keep trying out new possibilities, so as to

avoid premature convergence, but relevant past information has also to be transmitted

in the next generation to guide the behaviour of the algorithm.

In order to identify a set of good models, we set the parameters of the GA so as to

have fast convergence to local optima. This is achieved by slightly enforcing the

exploitation capabilities of the GA. Clearly, in this case it is fundamental to guarantee

that an adequate degree of exploration is maintained. This is obtained by properly

setting the overall stopping rule, which determines the number of times that the

genetic algorithm is independently applied to different initial populations.

The algorithm starts with the random generation of a population of individuals at

time $t=1$. After the fitness value is computed in correspondence of each individual,

the 5% of the best individuals are retained and directly inserted in the population of

generation $t+1$ (elitism). Then, using a linear fitness ranking assignment mechanism

and a stochastic universal sampling selection method (Baker, 1987), the individuals

are selected to constitute the intermediate population, which will be processed

through the crossover and the mutation $(p_m = 0.7/K)$ operators in order to determine the population of the generation at $t+1$. The genetic algorithm stops when the best individual of the population does not change for ten consecutive generations.

The crossover operator is the core mechanism of the genetic algorithm because it allows the transmission and recombination of relevant information across generations. Since several iterations of the GA are considered, it makes sense to improve the GA explorative capabilities by using two different operators. Here, we use both a standard uniform crossover operator (*XEdge*) and an *ad hoc* crossover operator based on the graphical structure of the model (*XVertex*). For any pair G^1 and G^2 of parents, XEdge randomly exchanges bits, i.e. edges. Edges have fixed independent probability p_c^E of being exchanged. XVertex consists in choosing randomly a subset A of V and then exchanging the subgraphs G_A^1 and G_A^2 to form two offspring (Poli and Roverato, 1998). Vertices have fixed independent probability p_c^V of belonging to A. We set p_c^E equal to 0.7, while to have significant comparison, p_c^V is such that

$$p_c^V = \binom{\text{int}(p_c^A \times |V|)}{2} / K, \text{ where int}(x) \text{ denotes the integer part of } x.$$

We randomly generate an initial population and, in turn, apply the GA-XEdge and GA-XVertex. These lead to the identification of two, possibly identical, models to be included in H. The procedure is iterated $N/2$ times. N is not set a priori. It has to be larger than a given constant T, but the procedure is iterated until no new solution is identified in the last R repetitions of the GA. This constraint guarantees that the identified solutions are the result of an adequate exploration of the search space because they are repeatedly identified in several runs of the GA with different initial

starting points and with different crossover operators. Indeed, the convergence requirement is such that convergence is not attained if the number of different identified solutions increases along with the number of iterations of the GA.

4. Simulations

In this paragraph we present the application of the proposed procedure to simulated datasets. A set V of 10 binary variables is considered, so that the search space is made up of $2^{45}>10^{13.5}$ models. Previous analysis (see Poli and Roverato, 1998) showed that model search is more challenging when the underlying graph is sparse; that is the number of edges is low. For this reason we consider a graph G^* with 13 edges and cliques *{{aj}, {bceg}, {bf}, {ci}, {deg}, {dh}, {fj}}* and ten datasets of size n, randomly generated from a multinomial distribution following the conditional independence structure encoded by G^*. To avoid problems deriving from the use of sparse tables, we set $n=20,000$.

The evolutionary procedure was applied to all datasets on populations of 150 individuals by setting $T=40$ and $R=10$. In all cases convergence was reached for $N=40$ and, as a result of the simulations, we obtained the sets H_i for $i=1,…,10$. As reported in Table 1, the cardinality of the sets H_i varies from a minimum of 2 for H_9 to a maximum of 7 for H_7. Considering the ten datasets jointly, 36 models were identified, 23 of which obtained by both the GA-XVertex and the GA-XEdge.

Table 1: Number of models identified by the procedure for the ten datasets.

DATASET	$|H_i|$=Total Number of Models Identified	Number of Models Identified by GA-XVertex	Number of Models Identified by GA-XEdge
1	3	2	3
2	3	3	2
3	3	2	3
4	5	3	4
5	2	1	2
6	3	3	2
7	7	4	6
8	5	5	4
9	2	2	2
10	3	3	3
TOTAL	36	28	31

The number of edges in the selected models varies from a minimum value of 12 to a maximum of 23. All the selected models have turned out to be local optima, that is having better fitness value with respect to their neighbour models differing exactly by one edge. Furthermore, all the selected models can be accepted on the basis of the generalised likelihood ratio test against the saturated model.

Note that, the combined use of two different crossover operators clearly enhances the explorative capability of the procedure without making its convergence slower since, as shown in Table 2, GA-XVertex and GA-XEdge have comparable convergence time.

Table 2: Summary statistics for the number of generations over 200 simulations of both GA-XVertex and GA-XEdge.

	Number of Generations				
	Minimum	**Mean**	**Max**	**Mode**	**Median**
GA-XVertex	26	36.46	57	34	36
GA-XEdge	20	36.38	66	34	35

It is of interest, at this point, to analyse the properties of the sets H_i. We first compare our procedure with a more traditional application of GA in model selection. For this reason we modified the parameters of the GA so as to make the convergence slower and identify, in a consistent way, a single model. We obtained that the model so selected was the fittest in H_i for all the 10 datasets. The sets H_i are thus made up of one global optimal model, typically identifiable by means of more traditional GA procedures, and of a few suboptimal models. Such suboptimal models represent the novel contribution of the proposed procedure, and we show that they provide relevant information and that, in fact, there is no reason why, even in a scientific modelling approach, the fittest model H_i in should be preferred to the remaining models in H_i.

For $i=1,…,10$, the set H_i is made up of J_i different models, and we denote them by the corresponding independence graph G_{ij}, $j=1,…, J_i$. The models in H_i can be ranked according to their fitness value and we give rank 1 to the model with best fitness value. Furthermore, as a measure of G_{ij} from G^* we consider the number of edges to be switched in G_{ij} to obtain G^*, and rank the models in H_i giving rank 1 to the model with graph closest to G^*. In this way we associate to each model two values (given in Table 3) representing the relative goodness of the model with respect to the other models in H_i. The strength of the association between such ranking scores can be

regarded as an indicator of the usefulness of the fitness function in discriminating the models in H_i. Conversely, the absence of interaction between the two scores denotes that, within H_i, the value of the fitness function is not a reliable measure of goodness of fit. Since the Fisher's exact test applied to Table 3 gives a p-value of 0.16 supporting the hypothesis of independence of the two ranking scores, we can conclude that all the models in H_i should be considered on an equal footing. In a technological modelling approach the models in H_i *can* co-exist, whereas in a scientific modelling approach, context-dependent considerations should be used to choose one of the models in H_i.

Table 3: Fitness function score vs Error score for the 36 selected models.

	Model	Fitness Score					
		1	**2**	**3**	**4**	**>= 5**	
	1	4	3	6	0	1	14
Error Score	**2**	1	4	1	1	1	8
	3	3	2	1	0	1	7
	4	1	1	0	0	0	2
	>=5	1	0	0	2	2	5
		10	10	8	3	5	36

The proposed automatic procedure has been implemented by using MATLAB and CoCo softwares: the procedure is written in MATLAB language. The MATLAB code calls CoCo (Badsberg, 1995) and exploits its efficient algorithms for the estimation of the log-linear models and the evaluation of the fitness function.

5. Applications to real data

In this section we apply the proposed procedure to the analysis of two real datasets, collected from economical and socio-medical environment.

5.1 A socio-medical problem

We first apply our procedure to a dataset collected within a socio-medical context (Badsberg 1995, p. 14). The dataset[1] has 1082 observations for 9 variables:

a)   Force ejection volume – lung function (<177, =177),

b)   cholesterol (<675, = 675),

c)   hypertension (yes, no),

d)   body mass index (normal 20.2-24.9, moderately obese 25.0-29.9, severe obese >30.0),

e)   smoking (yes , no),

f)   alcoholic consumption (rare, frequent),

g)   work (yes, no),

h)   gender (male, female),

i)   cohort-survey year (1967, 1984).

The procedure converged for $N=40$ and led to the identification of 5 models, which correspond to local optima with respect to their neighbour models and can all be accepted on the basis of the generalized likelihood ratio against the complete model.

---

[1] The dataset can be downloaded from the website www.math.auc.dk/~jhb/CoCo/Examples/datasets/

The edges of the independence graphs corresponding to the five selected models are given in Table 4. The minimum number of edges is 21 and the maximum 23. 18 edges (highlighted in Table 3) are common to all models; note that no one of the models is a subset of any of the others. From a comparison of the five models it turns out that, apart from edge *(e,f)* that belongs to four out of five graphs, all the differences involve variable c. This is graphically displayed in Figure 2 where the five independence graphs are jointly represented with dashed lines corresponding to edges that are not common to all graphs.

In this example, the selection of a set of models allows to clarify that for this dataset inference concerning the association between variable c and the remaining variables requires special attention. Clearly, any automatic search procedure designed to select exactly one model would miss to evidentiate such a feature of the data and would therefore not be appropriate for this problem.

In a scientific modelling approach only one model has to be chosen and, in absence of expert opinion to help in the selection strategy, a sensible choice could be to consider the unique model largest than the five selected models. Indeed, the five selected models are all submodels of such a largest model whose graph, in Figure 2, has just three edges more then the graph of model 3.

Table 4: Five selected model for the socio-medical dataset.

Model 1	Model 2	Model 3	Model 4	Model 5
	(a,c)	(a,c)		(a,c)
(a,d)	(a,d)	(a,d)	(a,d)	(a,d)
(a,e)	(a,e)	(a,e)	(a,e)	(a,e)
(a,f)	(a,f)	(a,f)	(a,f)	(a,f)
(a,g)	(a,g)	(a,g)	(a,g)	(a,g)
(a,h)	(a,h)	(a,h)	(a,h)	(a,h)
(a,i)	(a,i)	(a,i)	(a,i)	(a,i)
(b,c)		(b,c)		(b,c)
(b,d)	(b,d)	(b,d)	(b,d)	(b,d)
(b,h)	(b,h)	(b,h)	(b,h)	(b,h)
(b,i)	(b,i)	(b,i)	(b,i)	(b,i)
(c,d)	(c,d)	(c,d)	(c,d)	(c,d)
			(c,e)	
		(c,f)		
	(c,g)	(c,g)		(c,g)
(c,h)				(c,h)
(c,i)	(c,i)		(c,i)	
(d,e)	(d,e)	(d,e)	(d,e)	(d,e)
(e,f)	(e,f)	(e,f)	(e,f)	
(e,h)	(e,h)	(e,h)	(e,h)	(e,h)
(e,i)	(e,i)	(e,i)	(e,i)	(e,i)
(f,h)	(f,h)	(f,h)	(f,h)	(f,h)
(f,i)	(f,i)	(f,i)	(f,i)	(f,i)
(g,h)	(g,h)	(g,h)	(g,h)	(g,h)
(g,i)	(g,i)	(g,i)	(g,i)	(g,i)
(h,i)	(h,i)	(h,i)	(h,i)	(h,i)
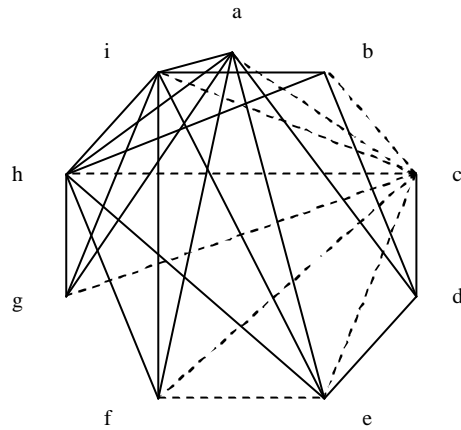
Figure 2: Joint representation of the independence graphs for the five selected models.

5.2 A credit scoring problem

Statistical credit scoring is a class of statistical method to determine the probability that a money borrower will repay on time the amount of credit she/he is granted. For a description of the problem see Hand and Henley (1997). In a Bayesian approach to inference, Tarantola, Giudici and Green (2000) fitted a graphical model to a set of 9 binary variables measured on 1000 borrowers:

a)   Gender (female, male),

b)   marital status (married-divorced-widowed, single),

c)   balance of current bank account (null-negative, positive),

d)   values of savings or stocks (= 500 DM, >500 DM),

e)   payments of previous credits (problematic, otherwise),

f)   credit amount (= 5000 DM, >500 DM),

g)   purpose of the credit (professional, otherwise),

h)   duration (= 24 months, >24 months),

i)   creditability (yes, no).

See also Fahmeir and Hamerle (1984) and Fahrmeir and Tutz (1994) for previous analysis of these data[2].

In this case our procedure led consistently to the identification of one single model, i.e. $|H|=1$, with cliques *{{ab},{afg},{ah},{cdi},{cf},{dei},{egi},{ehi},{bi}}*. This model is strongly supported by the data and consistent with previous analyses.

It may seem that there is no real need to use our procedure for this problem because any adequately tuned GA, designed to select one single model, would have led to the same result. However, it is worth pointing out that the use of our procedure allows to exclude the existence of relevant suboptimal models and enforces our confidence on the goodness of the selected model.

6. Discussion

The availability of efficient computational resources and of modern search heuristic algorithms has made possible an increasingly more effective exploration of the search space. Here, we have shown that an automatic search procedure can be designed to retain wider information on the structure of the search space than just the optimal value, thereby allowing a more comprehensive understanding of the model selection problem.

---

[2]   The dataset can be downloaded from the website www.stat.uni-muenchen.de/Einrichtungen/Rechnerbetrieb/Data-sets/Kredit/kredit-e.

A related problem concerns the role of the fitness function. The identification of exactly one, fittest model, is based on the implicit assumption that there is a monotonically increasing functional relation between the fitness value of a model and the closeness of such model to the "true" generating process. As shown in Section 4, this assumption can be hardly justified. A procedure that selects several models, as the one proposed here, is clearly less affected by the lack of validity of such an assumption and therefore provides more robust answers.

References

Baker, J., 1987. Reducing bias and inefficiency in the selection algorithm. Genetic Algorithm and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, ed J.J. Grefenstett.

Badsberg, J.H., 1995. A guide to CoCo. (Technical Report 91-43, Institute for Electronic Systems, University of Aalborg).

Bargelt, C. and Kruse, R., 2002. Graphical Models: Methods for Data Analysis and Mining. Wiley: Chichster.

Blauth, A. and Pigeot, I., 2000. GraphFitI - A computer program for graphical chain models.

Chatterjee, S., Laudato, M., Lynch, L.A., 1996. Genetic Algorithm and their statistical applications: an introduction. Computational Statistics & Data Analysis, 22,633-651.

Chickering, D.M., 1996. Learning Bayesian networks is NP-Complete. In Fisher, D. And Lenz, H., editors, Learning from Data: Artificial Intelligence and Statistics V, Springer-Verlag, 121-130.

Christensen, R., 1997. Log-linear models and Logistic regression. 2nd Ed., Springer.

Cowell, R.G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J., 1999. Probabilistic Networks and Expert Systems, Springer-Verlag, New York.

Darroch, J.N., Lauritzen, S.L., Speed, T.P., 1980. Markov fields and log-linear interaction models for contingency tables. Ann.Statist., 11, 3, 724-738.

Edwards, D.E., 2000. Introduction to Graphical Modelling, Springer-Verlag, New York.

Exteberria, R., Larranaga, P., Picaza, J.M., 1997. Analysis of the behaviour of genetic algorithms when learning Bayesian network structure from data. Technical report.

Fouskakis, D., Draper, D., 2002. Stochastic optimization: a review accepted for publication in Journal of the International Statistical Review.

Goldberg, D.E., 1989. Genetic algorithms in search, optimization and machine learning, Reading, Addison-Wesley Publishing Corporation, Inc.

Hand, D.J. and Henley, W.E., 1997. Statistical methods for consumer credit scoring: a review. J.R.Stat. Sci. A, 160, 523-541.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Harbor.

Larranaga, P., Etxeberria, R., Lozano, J.A., Sierra, B., Inza, I., Pena, J.M., 1999. A review of the cooperation between evolutionary computation and probabilistic graphical models, Proceedings of the II Symposium on Artificial Intelligence:314-324, CAMAF 99, La Habana, Cuba.

Lauritzen, S.L., 1996. Graphical Models, Oxford: Oxford University Press.

Mitchell, M., 1996. An introduction to genetic algorithms, MIT press.

Poli, I., Roverato, A., 1998. A genetic algorithm for graphical model selection, Journal of Italian Statistical Society, Vol.7, No.2.

Tarantola, C., Giudici, P., Green, P., 2000. Efficient Model Determination for Discrete Graphical Models, Technical Report, University of Pavia, Italy.

Whittaker, J., 1990. Graphical model in Applied Multivariate Statistics. John Wiley and Sons, Chichester.