Massimiliano Mascherini

Department of Statistics “G.Parenti”
University of Florence, Italy

MASTINO is built on the top of the package DEAL, developed by S. G. Bøttcher and C. Dethlefsen (2003).

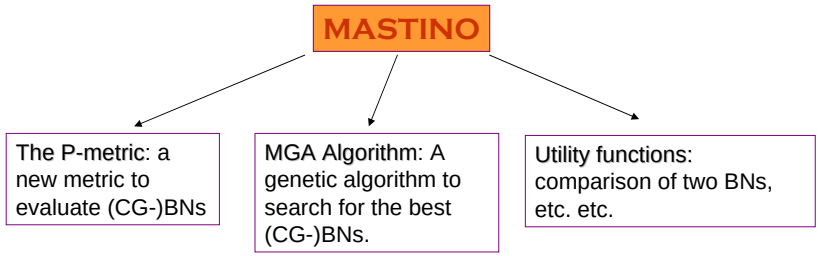
Using data structures and functions already implemented in DEAL, (i.e. network definition, prior on parameter, BDe metric, etc. etc.), MASTINO provides various functions and methods enhancing the learning of Conditional Gaussian Bayesian Networks from data.

MASTINO is a suite of R functions to learn Bayesian Networks from data

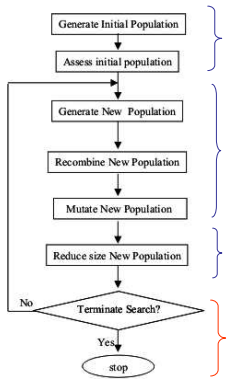
It is born by the implementation of the new methods for learning BNs from data I proposed in my PhD thesis, entitled “Learning Probabilistic Networks in Large and Structured Domains” and successfully defended last February at the Department of Statistics of the University of Florence, Italy.

MASTINO is freely downloadable as collection of R functions from my website:
www.ds.unifi.it/mascherini

In particular, in MASTINO the **P-metric**, (M.Mascherini and F.M. Stefanini, 2005), a new original metric to evaluate Bayesian Networks using prior information on structures, and the **MGA algorithm**, (M.Mascherini and F.M. Stefanini, 2005), a genetic algorithm to search for the best Conditional Gaussian Bayesian Networks, are implemented as well as numerous others utility functions to manage with BNs.



The **MGA algorithm** is a population-based heuristic strategy to search for the best conditional gaussian bayesian networks maximizing the BDe metrics, extended for CG-BNs by S. G. Bøttcher (2005).



1) Individuals (CG-Bayesian Networks) are randomly generated!

2) Individuals reproduce themselves in the Offspring production process

3) According to a given metric, Individuals are selected and die!

Steps 2 and 3 are iterated until the stop condition is reached

A random population of size *K* of BNs is created, where each BN is created using methods implemented in DEAL.

Then, following the Larranaga (1996) approach, each Bayesian Network B_i is represented as a Connectivity Matrix *CM* of dimension (n by n), where each element, c_{ij} , satisfies: c_{ij} = (1 if j is a parent of i, 0 otherwise).

$$CM(B_i) = \begin{bmatrix} c_{i\to i} & c_{i\to j} & \dots & \dots & c_{i\to k} \\ c_{j\to i} & c_{j\to j} & \dots & \dots & c_{j\to i} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ c_{k\to i} & c_{k\to j} & \dots & \dots & c_{k\to k} \end{bmatrix}$$

Then, the Connectivity Matrices are linearized in individuals (Connectivity Vectors) $CV(B_i)=(c_{11},c_{12},c_{13},\dots,c_{nn})$, and the population of K strings will be the starting population of genetic algorithm.

$$CV(B_i) = [c_{i\to i}, c_{i\to j}, \dots, c_{i\to k}, c_{j\to i}, \dots, c_{j\to k}, \dots, c_{k\to i}, c_{k\to j}, \dots, c_{k\to k}]$$

When all the random BNs are coded as Connectivity Vectors, the Offspring Production process can start and new individuals are randomly created. Although the crossover and the mutational operators are maintained, the Offspring Production process here adopted quite differ by the process implemented by Larranaga: it improves the genetic variability of the population that permits to reach a faster convergence and avoiding local maxima.

$$CV(B^1) = [I_{i\to i}, \dots, I_{i\to k}, I_{j\to i}, \dots, I_{j\to k}, \dots, I_{k\to i}, I_{k\to j}, \dots, I_{k\to k}]$$
$$CV(B^2) = [I_{i\to i}, \dots, I_{i\to k}, I_{j\to i}, \dots, I_{j\to k}, \dots, I_{k\to i}, I_{k\to j}, \dots, I_{k\to k}]$$

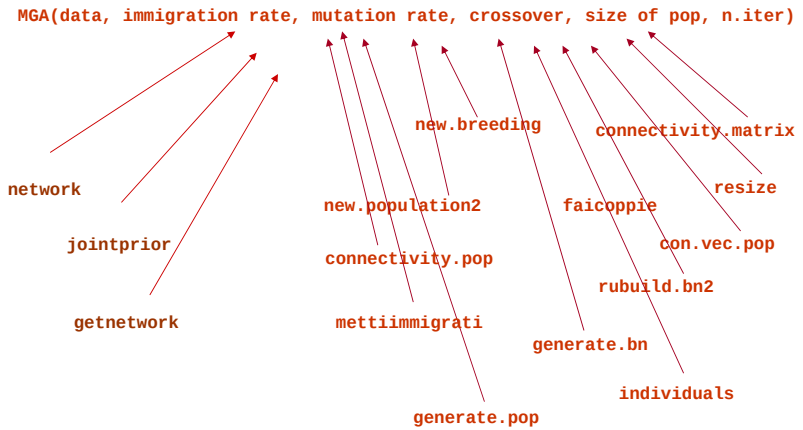
If $I_{i\to j} = I_{i\to j}$ then $I^{NEW}_{i\to j} = I_{i\to j}$

Else the new individual iinherits as follow: $P(I^{NEW}_{i\to j} = I_{i\to j}) = p$; $P(I^{NEW}_{i\to j} = I_{i\to j}) = 1-p$

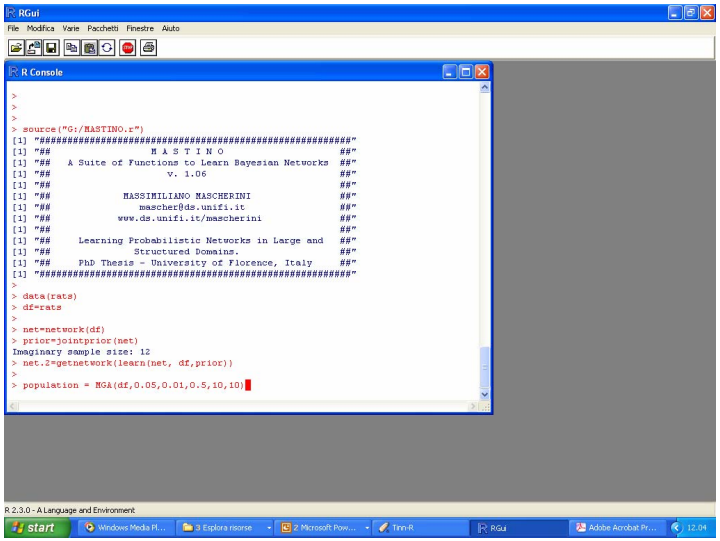
The admissibility of the structure entailed by the new individual is checked by testing the fulfilment of CG-BN properties and DAG requirements. A random elimination of inadmissible arcs is performed if the test fails.

Then population is resized to the original size following the elitist criterion

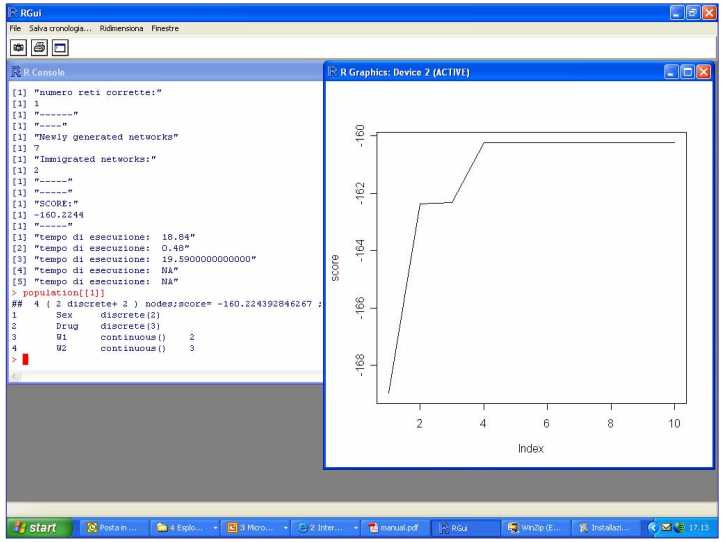
The entire algorithm is coded as an R function, composed by many other R functions…..



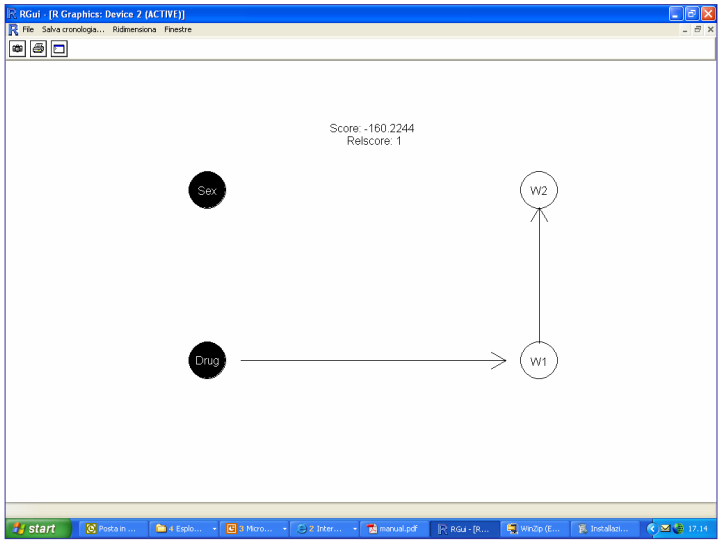
A simple example!



A simple example!

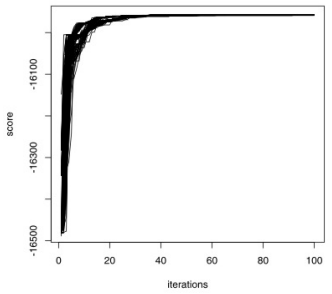


A simple example!



We tested the algorithm with several Machine Learning benchmark dataset with successful results

Using the KSL case study, Badsberg (1995), included in DEAL, the convergence to the real network is achieved after an average number of iterations equal to 84.57.



In general, the number of iterations required for the convergence is lower than other genetic algorithm approaches (Larranaga, 1996).

Obviously, the time of computation depends by the size of the networks, it varies from few seconds (rats data) to one hour (ML benchmark datasets) or more.

The **P-metric** is a new metric to evaluate BN exploiting prior information on structures that the expert can have.

Most of the approaches developed in the literature to elicit the a-priori distribution on BNs structures require a full specification of graphs, (Buntine, 1991; Heckerman 1994). Unfortunately, an expert can have more knowledge about one part of the domain than another making unfeasible a coherent complete specification of the prior structure distribution

We proposed a prior elicitation procedure for DAGs which exploits weak prior knowledge on DAG's structure and on network topology.

$$S_{Prior}(B_s) = S^{\partial}_{Prior}(B_s) + S^{\tau}_{Prior}(B_s)$$

the first part encodes the elicitation of the belief over arcs, when available.

the second part encodes the belief the expert has over the resulting topology of the candidate structure.

Defined the prior elicitation procedures, we then develop a new quasi-Bayesian score function, the ***P-metric***, to evaluate BN and to perform structural learning following a score-and-search approach

$$S_{P-metric}(B) = S_{prior}(B) \cdot P(D \mid B, \xi)$$

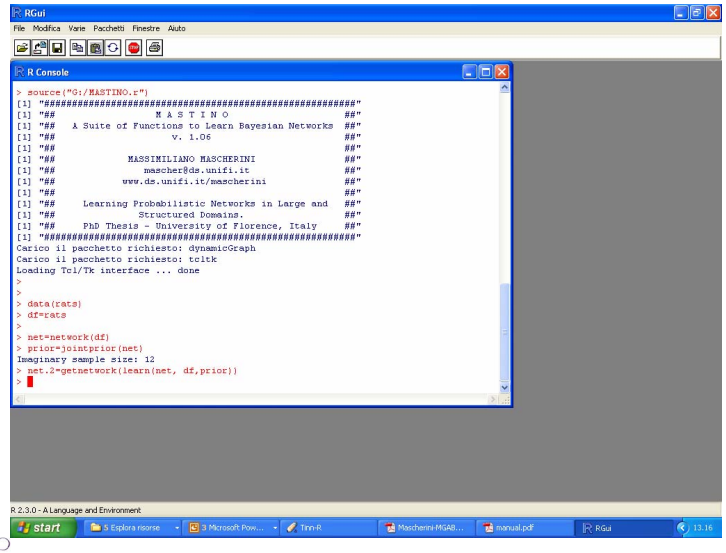
BDe likelihood

In MASTINO we have implemented functions to encode the prior elicitation procedures and the P-metric, using Greedy Search heuristic strategy to find the best BNs.

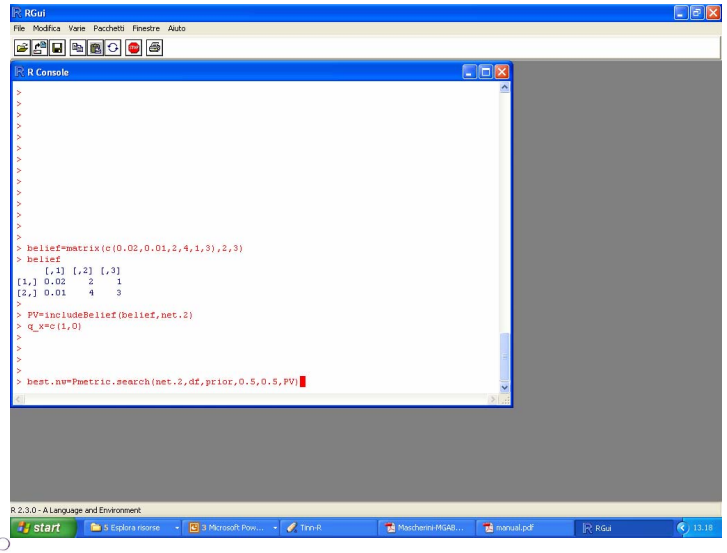
```
Pmetric(network , alpha, class bound, ProbabilityVector)
```

We tested this new methods with several ML benchmark dataset with successful results, even if, for more complex databases we faced the computational limitations of the R environment and some strange behaviour of the BDe metric implemented in DEAL.

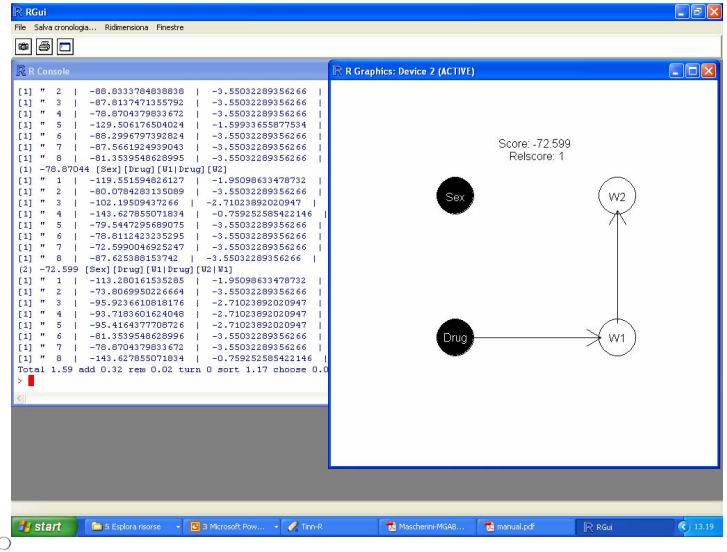
A simple example!



A simple example!



A simple example!



THE PROBLEMS!

Computational Burden

We tested our algorithms using a IBM e-server, a dual processor computer equipped equipped with 2xAMD Opteron 2.0GHz (1MB L2 Cache) with 5giga RAMs and the operative system is Red Hat Enterpriser Linux AS Ver.4.

Running several tests with ML benchmark datasets we found that the out of memory error is often invoked.

The Out of Memory error arises when dealing with networks handling more than 27 (discrete) nodes and/or when using large sample space.

Larger sample size → Lower size of the network supported

THE PROBLEMS!

Computational Burden

DEAL is afflicted by the same problem! We found that the problem arises especially during the **`sort`** phase of the greedy search.

A question naturally arises:

It is R the perfect environment when dealing with problem with this size of complexity???

Who knows?? I just know that at the moment large networks are untreatable using DEAL or MASTINO…

THE PROBLEMS!

BDe metric implemented in DEAL

Testing our package MASTINO, we massively used the package DEAL.

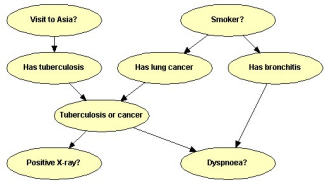
During my tests, I've found that often the learned network converges towards a complete networks and not to the true network

Checking the results, it seems that the BDe metric implemented in DEAL appears to be greatly dependant by the imaginary sample size of the jointprior function.

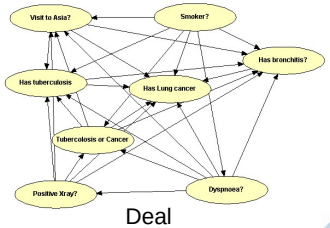
THE PROBLEMS!

ASIA Network (Lauritzen et al., 1988)

Sample size	Total number of arcs	Correct Arcs	Wrong Directed Arcs	Incorrect Added Arcs	Missing Arcs
500	27	2/8	6	19	0
1500	26	1/8	7	18	0
3000	26	1/8	7	18	0
5000	26	1/8	7	18	0



Original



Deal

THE PROBLEMS!

HGH Network reduced to 20 variables, (Le et al., 2005)

Sample size	Total number of arcs	Correct Arcs	Wrong Directed Arcs	Incorrect Added Arcs	Missing Arcs
500*	48	1/33	18	29	14
1500**	40	1/33	18	21	14
3000***	19	0/33	7	12	26
5000****	10	0/33	5	5	33

* Stopped by out of memory error after 49 iterations

** Stopped by out of memory error after 40 iterations

*** Stopped by out of memory error after 19 iterations

**** Stopped by out of memory error after 10 iterations

Conclusion!

Here we have presented the package MASTINO, to learning BNs from data.

MASTINO is build on the top of the package DEAL and provides various methods to learn BNs from data

MASTINO was tested with several Machine Learning benchmark dataset with successfully results.

Unfortunately there is a strong limitation: for real networks the problem of the computational burden arises, making unfeasible the process of structural learning.

Bibliography!

- J. H.Badsberg – *An environment for Graphical Models* – PhD thesis, Aalborg University – Denmark – 1995
- S.G. Bøttcher, - *Learning Conditional Gaussian Networks*, PhD Thesis, Aalborg Universitet - 2005
- S.G.Bottcher and C. Dethlefsen – *DEAL: A package for learning Bayesian Networks* – AAU.DK technical report – 2003
- P. Larranaga and M.Poza – *Structure Learning of Bayesian Networks by genetic algorithm: a performance analysis of control parameters* - IEEE Journal on Pattern Analysis and Machine Intelligence, 1996
- S.L. Lauritzen and D.J. Spiegehalter - *Local Computation with probabilities on graphical structures and their application to expert system,* JRSS, 50(2):157-192, 1988.
- M. Mascherini and F.M. Stefanini, *Encoding structural prior information to learn large bayesian networks*, WP 2005/13, Florence University Press
- M. Mascherini and F.M. Stefanini, *A Genetic Algorithm to Search for the Best Conditional Gaussian Bayesian Network* – Proceedings of the IEEE International Conference on Computational Intelligence for Modelling, Control and Automation, IEEE Computational Intelligence Society, U.S.A.
- P.P. Le and A. Bahl and L.H. Ungar - *Using prior knowledge to improve genetic network reconstruction from microarray data,* InSilico Biology, 27(4), 2004