MASTINO:
Learning Bayesian Networks Using R

Massimiliano Mascherini[1], Fabio Frascati[2], and Federico M. Stefanini[2]

[1] European Commission, Joint Research Centre
Via E. Fermi 2479, 21027 Ispra(Va), Italy, *massimiliano.mascherini@jrc.it*
[2] Dipartimento di Statistica 'G.Parenti', University of Florence
V.le Morgagni 59, 50100 Florence, Italy, *stefanini@ds.unifi.it*
fabiofrascati@yahoo.it

Abstract. Bayesian Networks are increasingly used to represent conditional independence relations among variables and causal information in problem domains in which decisions are based on probabilistic reasoning. Structural learning is NP-hard therefore the database of observed cases must be often supplemented with search heuristics based on prior information. In this paper we present a software package for R, called MASTINO, that extends the existing DEAL package by providing new tools for learning Bayesian Networks and Conditional Gaussian networks in a score-and-search framework, such as the score function *P-metric* and the *M-GA* genetic algorithm. MASTINO is freely available under the terms of the GNU General Public License Version 2, and it has been recently submitted to be part of the CRAN repository. Meanwhile it can be downloaded from the website: `http://statind.jrc.it/mastino`.

Keywords: Bayesian Networks, Structural Learning, R Package

1 Introduction

Bayesian Networks (BNs), Cowell et al. (1999), are a widespread tool in many areas of artificial intelligence and statistics because of efficient algorithms which make probabilistic inference effective in highly structured problem domains. BNs are suited to represent conditional independence relationships but they have been extended to represent causal information, Spirtes et al. (2000), and utility of decisions, so that probabilistic expert systems are increasingly developed in areas ranging from technology to medical problem domains.

Inference about the structure of a BN, also called structural learning, has been proved to be a NP-hard problem, Chickering (1995). Structural learning is typically performed by combining expert's priori knowledge with the information contained in a database of cases. Several heuristics have been shown to work in practice and it seems that specialized problem domains take benefits from problem-dependent tuning. A software package for `R`, (R Development Core Team, 2008), suited to quickly implement hypothesized heuristics and

to test them against standard benchmarks in structural learning is therefore welcome.

Structural learning of Bayesian Networks has been first implemented in `R` by Bøttcher and Dethlefsen (2003), who wrote the DEAL package in which the BDe metric was implemented for learning Conditional Gaussian (CG) networks, Bøttcher (2005). A Greedy Search algorithm with random restart is the search engine optimizing the BDe score.

In this paper we present MASTINO, a `R` package that provides new tools for learning BNs and CG networks. MASTINO extends DEAL in several ways. In particular, the P-metric score function is implemented to evaluate CG newtorks under strong but partial prior information on the structure, Mascherini and Stefanini (2005b, 2007). The *M-GA* genetic algorithm is based on a new population-based heuristic aimed at a robust search for the best CG network, Mascherini and Stefanini (2005). The suite of functions includes some utilities to help with the manipulation of BNs and CG networks. MASTINO is freely available under the terms of the GNU General Public License Version 2, and it can be downloaded from the website: `http://statind.jrc.it/mastino`. The package works under a `R` version >= 2.4.1 and it has been recently submitted to be part of the CRAN repository, therefore it should be soon available for download among contributed packages.

This paper starts with Section 2 in which Bayesian Networks are shortly described and structural learning of Bayesian Networks is introduced in the score-and-search approach. New methods implemented in MASTINO are explained. Then, Sections 3 and 3 describe two simple examples together with the corresponding `R` code and conclusions and issues to be addressed by further research concludes the paper. For the discussion of more complex real data examples we address the reader to Mascherini and Stefanini (2007).

2 Learning Bayesian Networks

A BN is a graph-based representation of random variables encoding their joint probability distribution in a compact way. For terminology and theoretical aspects on BNs, we refer to Cowell et al. (1999). In this paper discrete BNs are shortly defined as a directed acyclic graph (DAG) $D = (V, E)$ where V is a finite set of vertices and E is a finite set of directed edges between vertices. The DAG D encodes the structure of the Bayesian Networks. To each vertex $v \in V$ in the graph corresponds a discrete random variable X_v. The set of variables associated with the graph D is $X = (X_v)_{v \in V}$. For shortness, sometimes the label v also indicates the correspondent random variable X_v. In addition, for each vertex v a set of parents, $pa(v)$, is defined. A conditional probability table (CPT) is attached to every pair $(v, pa(v))$. Thus the set P of all local probability distributions $p(x_v \mid x_{pa(v)})$ is obtained. The joint

probability distribution is defined through the factorization:

$$p(x) = \prod_{v \in V} p(x_v \mid x_{pa(v)})$$

In order to completely specify a Bayesian Network for X, we must therefore specify a DAG D and a set P of local probability distributions, operatively CPTs. Note that nodes at the tail of directed edges reaching node v denote the random variables in the conditional distribution of X_v, thus a conditional independence assertion is associated to the lack of one or more directed edges. Further conditional independence relations may be read from the graph by exploiting separation theorems. CG-BNs are probabilistic Networks in which both continuous and discrete random variables are present. To ensure exact local computation, discrete random variables are not allowed in CG-BNs to have continuous parents. A method to perform parameter and structural learning in CG-BNs has been recently described in Bøttcher (2005), where a comprehensive discussion is performed.

The set of directed edges E on V defines a DAG, the structure of a BN. Structural learning of BNs may be performed following two main different approaches. In the first approach, learning follows the original PC schema, Spirtes et al. (2000), which performs statistical tests to produce a list of conditional independency relations. In the second approach, called "score-and-search", algorithms compare candidate DAG structures according to a given score, also called metric, which is used as objective function during optimization. Widely used scores include BIC, AIC and MDL. The most important score from the Bayesian standpoint is the BDe metric, Heckerman et al. (1995), in which a candidate DAG structure B_s on a fixed set of nodes V is supported by observed data if the conditional posterior distribution of B_s given observed data is large.

The above metrics have been all successfully used in actual learning tasks. Despite the recognized possibility of improving the learning process by exploiting prior information, attempts to elicit prior beliefs on networks structure in a quantitative way are still quite limited in the literature.

Mascherini and Stefanini (2005, 2007) proposed a new score function, the P-metric, which mixes prior beliefs and experimental information following the BDe assumptions, Heckerman et al. (1994). In particular, the P-metric encodes the a-priori belief on the structure of a candidate network B_s by a score function $S_{prior}(B_s)$ which captures some local and some global network features. Local features are defined by score component $S_p^{\delta}(B_s)$ that describe beliefs on the presence of oriented edges, each one marginally considered. Partial prior beliefs on network topology is encoded by score component $S_p^{\tau}(B_s)$, which takes the form of an expected degree of connectivity, for example the expected number of parents for one child, and it is scaled according to the Kullback-Leibler distance, (Kullback et al., 1951).

The proposed score function $S_{prior}(B_s)$ combines the two components $S_p^{\delta}(B_s)$ and $S_p^{\tau}(B_s)$ on the logarithmic scale:

$$S_{prior}(B_s) = \log\left[\left(\frac{P(B_s)}{P(\{\emptyset\})}\right)^{\alpha} \cdot e^{(1-\alpha)(-KL(\mathcal{P}_{pa}\|\mathcal{Q}_{pa}))}\right] \tag{1}$$

where the role of α, $0 \leq \alpha \leq 1$, is to balance the relative strength of components due to edge orientation and to network topology. A value $\alpha = 1$ is suited to perform learning without accounting for the network topology component. According to the prior belief, the most plausible structure is obtained by maximizing the score $S_{prior}(B_s)$ with respect to B_s.

Our Bayesian-inspired metric, called *P-metric*, mixes the elicited prior information and experimental information in a way close to Heckerman et al. (1994). The Bayesian Dirichlet with Equivalence metric, (BDe), assigns the same likelihood value to structures which are likelihood equivalent, i.e. DAGs encoding the same assertions on conditional independence relations. The equivalence is obtained by choosing BN parameters through a prior procedure in which Dirichlet hyperparameters are defined using the notion of equivalent sample size. Then, the *P*-metric defining the score of a candidate structure B_s given a complete database of cases $\mathcal{D}$ is, on the log scale:

$$log\left(S_{P\text{-}metric}(B_s)\right) = \beta_z \cdot \log(S_{prior}(B_s)) + ll_{BDe}(D \mid B_s, \theta) \tag{2}$$

The role of the parameter β_z is to calibrate the strength of the prior score with respect to the likelihood function. The value of β_z depends on the size of the problem domain and on the sample size of cases as well as on the elicited belief. Clearly for $\beta_z = 0$ the *P*-metric is equal to the BDe metric, if a uniform prior distribution over structures is chosen in the BDe. In MASTINO the P-metric is implemented with the `Pmetric()` function. The best network using the P-metric can be found by two heuristic strategies: the greedy search, `Pmetric.search()` and the perturbated hill-climb, `p.hill.climb()`, that are an extension of the algorithms already implemented in DEAL.

Focusing on the heuristic strategies, in order to search the best Conditional Gaussian Bayesian Networks using the BDe metric, in MASTINO a genetic algorithm, named *M-GA*, (Mascherini et al. (2005)), is implemented with the `MGA()` function.

Genetic algorithms (GAs) have been first used by Larrañaga et al. (1996) to search for optimal discrete BN structures. The GA implemented in MASTINO is a modification of the method proposed by Larrañaga et al. (1996) which also works with CG networks. In the *M-GA* the single crossover point of Larrañaga et al. (1996) is extended and the two parents of a new individual equally contribute to offsprings on a gene by gene basis, taking part bit by bit in the creation of the new individual string, to maintain or increase the genetic variability of the population of candidates. Furthermore, a fixed number of randomly generated networks is added at each generation as immigrants.

3 Examples

The ASIA network is a small fictitious and well-known Bayesian network, Lauritzen et al. (1988), for calculation of the probability of a patient having tuberculosis, lung cancer or bronchitis given values taken by some other variables, like visit-to-Asia which is one if the patient recently visited Asia. The problem domain is here quite rich, for example shortness-of-breath, called dyspnoea (D), may be due to different factors, like tuberculosis (T), lung cancer (L), and bronchitis (B). Then a recent visit to Asia, (A), increases the risk of tuberculosis, while smoking, (S), is known to be a risk factor for both lung cancer and bronchitis. Results of a single chest X-ray, (X), do not discriminate between lung cancer and tuberculosis, (E), as neither does the presence or absence of dyspnoea. All the 8 variables of the model are binary and the dataset included in MASTINO contains 1500 cases.

In MASTINO, the initialization of a network precedes the score-and-search step of structural learning. Being based on the package DEAL, the library MASTINO can be used to learn both discrete and CG networks. In particular MASTINO exploits the representation of a Bayesian Network as an object of class network defined in DEAL. Networks are generated from a dataframe, and discrete variables must be specified factors. In Bøttcher et al. (2003) a complete description of the DEAL functions is provided. Network building is performed using DEAL resources:

```
> library(MASTINO); data(asia); df = asia; net=network(df)
```

Parameter learning follows the Bayesian approach. Parameters of the joint distribution of variables in the network are determined by the function `newprior()`, that is based on the function `jointprior()` defined in DEAL. To improve the learning process fully discrete and CG networks are treated as different objects by setting automatically the parameters of the master prior function, Bøttcher (2003):

```
> prior=newprior(net); net.2=getnetwork(learn(net,df,prior))
```

After defining the prior distribution the P-metric may be used to learn the structure using expert's belief. We assume that the available prior information was partially quantified by experts concerning three pairs of nodes: (A, T), (S, L) and (L, T). The expert states that the node "Tuberculosis" (T) is not reputed to have any effect on "Visiting Asia" (A), so the probability of the event $A \leftarrow T$ was set to be equal to 0.01, and the remaining probabilities are set to capture a slight effect of the event "Visiting to Asia" on "Tuberculosis"; then, "Smoking" (S) was believed to have an effect on "Lung Cancer" (L) but "Lung Cancer" did not have any effect on "Smoking". The probability of those events was set to $P(S \rightarrow L) = 0.6$ and $P(S \leftarrow L) = 0.01$ respectively. Finally, no effects between "Lung Cancer" and "Tuberculosis" (T) was believed to exist, so the probability of the event $L \not\leftrightarrow T$ was set equal to 0.8. For simplicity, the prior information elicited above was then encoded in MASTINO using vectors, where the first value is equal to the prior

probability, the second is the identificative number (ID) of the parent node and the last is the ID of the child node:

```
> bel1=c(0.01,5,3); bel2=c(0.55,3,5); bel3=c(0.6,4,6)
> bel4=c(0.01,6,4); bel5=c(0.1,6,5); bel6=c(0.1,5,6)
```

The six vectors are then merged into a matrix and they are included in MASTINO using the dedicated function:

```
> belief=rbind(bel1,bel2,bel3,bel4, bel5, bel6)
> PV=includeBelief(belief, net)
```

Prior information on network topology was defined by requiring that 80% of network nodes has at most one parent, `q_x=c(0.8,0.2); cl=1`. The set of input parameters in the P-metric function must be specified before structural learning. In particular the strength of the prior information, β and the importance of the local features, α, must be specified. Mascherini and Stefanini (2007) numerically explored the effect of several different pairs of parameter values on the overall score. The P-metric seemed not highly sensitive to the precise numerical choice of two parameters. Based on these results, here we set $\beta = 0.5$, i.e the strength of the prior information is reduced of 50%, and α=0.75, because due to the small size of the network the local features are considered more important than the global features. Structural learning takes place by invoking `Pmetric.search` or `P.hill.climb`, which respectively extend the greedy search algorithm and the perturbated hill climbing implemented in DEAL:

```
> alpha= 0.75; beta=0.5; best.gs=
Pmetric.search(net.2, df, prior, beta, alpha, PV, cl, q_x)
> best.hc=p.hill.climb(net.2, df, prior, beta, alpha, PV, cl,
q_x)
```

DEAL provides functions to plot learned networks. In this case study the two algorithms converged to the same network presented in Figure (1, a). The comparison of these two networks is performed by means of function `compareBN`, which gives a summary of the similarities of the two networks in terms of correct/wrong arcs:

```
> plot(best.gs[[1]]); plot(best.hc[[1]])
> compareBN(best.gs[[1]],best.hc[[1]])
```

The comparison of the learned network with the original ASIA network shows that a total of 7 arcs (correct + reverse oriented) out of 8 are successfully identified by MASTINO. Although in the network learned using the P-metric one arc is missing, these findings suggest the overall effectiveness of our algorithms. In particular, the P-metric outperforms other algorithms as the BDE metric implemented in DEAL and PC-NPC algorithms implemented in the commercial software HUGIN, Andreassen et al. (1989). In fact, the P-metric correctly identified a total of 7 arcs in the best case against a total of 5 arcs discovered by the PC and NPC algorithms. The comparison of computer runs performed with the BDe metric implemented in DEAL shows an unexpected feature of the DEAL implementation: if prior parameters are automatically set by DEAL then the learning algorithm discover a total of 7 arcs out of 8 but it also adds other 17 incorrect arcs. On other hands, if prior

parameters are manually set then a total of 5 arcs are correctly identified by the BDe metric implemented in DEAL (Figure 1, b).

RATS is a network created by simulated data available in DEAL. The dataset is structured in 24 rats (12 females, 12 males) receiving a randomized assignment of one drug among three products for loosing weight. The weight loss for each rat is noted after one and two weeks. The variable included in the dataset are: Sex (discrete, binary), Drug (discrete, trinomial), W1 (numeric, weight loss, one week), W2 (numeric, weight loss, second week). The aim is to assess the effects of Drugs on the rats' weight loss. The network is initialized as described above:

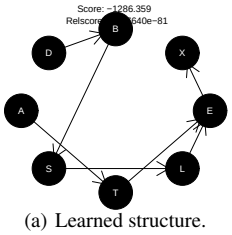
```
> data(rats); df = rats; net=network(df); prior=newprior(net)
> net.2=getnetwork(learn(net,df,prior))
```

After network initialization, structural learning of a CG network is performed by invoking the `MGA()` function, a population-based algorithm performing stochastic search in the space of CG networks. According to the literature, we set the following parameter values: immigration rate = 0.05; mutation rate = 0.01; crossover = 0.5; population size = 10; number of generations = 10. Besides setting different values for parameters of the M-GA algorithm, the user must specify at least a dataframe of observations: `>best.MGA = MGA(df,0.05,0.01,0.5,10,10)`. The 10 best structures from one run are contained into the list `best.MGA`. The graphical representation of the best structure is obtained by executing the function `plot`. The best network found by M-GA is equal to the original network and also to the network learned using the autosearch function implemented in DEAL. Values of the BDe score for top scored structures along generations are easily obtained and plotted as the output of the commands below shows:

```
>plot(best.MGA[[1]]); best.DEAL=autosearch(net.2, df, prior)
>compareBN(best.MGA[[1]], best.DEAL[[1]])
```

4 Conclusion

In this paper we presented MASTINO, a `R` package to learn Bayesian Networks following a score-and-search approach, which extends the DEAL pack-



(a) Learned structure.

Algorithm	Correct and Reverse Oriented	Missing Arcs	Incorrect Added
PC	5	3	0
NPC	5	3	1
BDe^{*}_{DEAL}	7	1	17
BDe^{**}_{DEAL}	5	3	2
P-metric	7	1	0

(b) *prior parameters automatically set; **prior parameters manually specified.

Fig. 1. Structural learning of the ASIA network in MASTINO [a]. Performances of some learning algorithms are compared at a sample size equal to 1500 [b].

age. The score metric, called *P-metric*, is implemented to evaluate structures using an informed score and the *M-GA* genetic algorithm is coded to perform a robust search in the space of CG networks. Although some computational constraints of `R` limit the use of MASTINO to problem domains with few variables, the package represents an original implementation of a set of recently proposed tools. Further work could be directed towards making MASTINO suited to learn large sized networks. A preliminary inquiry seems to suggest that a low-level recoding will both make MASTINO independent on DEAL and increase its speed up to handle reasonably large problems domains.

References

ANDREASSEN, S.K., OLESEN, K.G., JENSEN, F.V. and JENSEN, F. (1989): HUGIN: a shell for building bayesian belief universes for expert systems. *Proceedings of the 11th International Joint Conference on Artificial Intelligence.*

BØTTCHER, S.G. and DETHLEFSEN, C. (2003): DEAL: A package for learning bayesian networks. *Journal of Statistical Software 8(20), 1–40.*

CHICKERING, D.M. (1995): Learning bayesian networks is NP-complete. *Proceedings on Artificial Intelligence and Statistics, 121–130..*

COWELL, R.G., DAWID, P.A., LAURITZEN, S.L. and SPIEGELHALTER, D.J. (1999): *Probabilistic Networks and Expert Systems.* New York: Springer-Verlag.

HECKERMAN, D., GEIGER, D., and CHICKERING, D.M. (1994): Learning Bayesian Network: A combination of knowledge and statistical data. *Proceedings of 10th Conf. Uncertainty in Artificial Intelligence, 293–301.*

KULLBACK, S. AND LEIBLER, R. A. (1951): On information and sufficiency. *Annals of Mathematical Statistics 22, 79-86.*

LARRAÑAGA, P. and POZA, M. (1996): Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters. *IEEE Journal on Pattern Analysis and Machine Intelligence 18(9), 912-926.*

LAURITZEN, S. and SPIEGEHALTER, D. (1988): Local computation with probabilities on graphical structures and their application to expert system. *Journal of the Royal Statistical Society - B Series 50(2), 157–192.*

MASCHERINI, M and STEFANINI, F.M. (2005): M-GA: A genetic algorithm to learn Conditional Gaussian Bayesian Networks. *Proceedings of the IEEE International Conference on Computational Intelligence for Modelling, Control and Automation.*

MASCHERINI, M and STEFANINI, F.M. (2005b): Encode prior information to learn bayesian networks. *WP n.13 of the Department of Statistics*, Florence University Press.

MASCHERINI, M and STEFANINI, F.M. (2007): Using Weak Prior Information on Structures to Learn Bayesian Networks. *Lecture Notes in Artificial Intelligence. 4692(1), 413–420.*

R DEVELOPMENT CORE TEAM (2008): *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051- 07-0, URL http://www.R-project.org.

SPIRTES, P., GLYMOUR, C. and SCHEINES, R. (2000): *Causation, Prediction, and Search, 2nd ed.* New York, N.Y.: MIT Press.