LETTERS IN BIOMATHEMATICS
(URL to be inserted upon acceptance)

Strategies to avoid overfitting of MCMC Bayesian learning in some biological applications

Diego Garcia[a], Irene Tischer[a]

[a]School of Systems and Computing Engineering, Universidad del Valle, Santiago de Cali, Colombia

ABSTRACT

Model learning from observed data is typically affected by overfitting, because in order to find the models best parameter set, all relations between data are used indifferently whether they represent relevant or noisy interactions. Bayesian networks are widely used in biological modeling (e.g. networks of gene interactions), given that they allow representing graphically and determining statistically the dependence /independence relations between considered variables. A frequent approach in Bayesian learning is Markov Chain Monte Carlo simulation (MCMC), where a set of viable networks are explored by a random walk which converges to a network fitted optimally to data with respect to the likelihood or similar evaluation function. Here we propose various strategies to mitigate overfitting in Bayesian learning by MCMC in order to reduce the resulting models' complexity. They either apply constraints inside the MCMC simulation or consider post-optimal operations. We show the effectiveness of these strategies in some biological applications.

ARTICLE TYPE
Research Article

ARTICLE HISTORY
To Be Determined
To Be Determined

KEYWORDS
Bayesian networks, Bayesian learning, MCMC simulation, overfitting

1 Introduction

Bayesian networks are a powerful tool of knowledge representation and reasoning under uncertainty conditions, that often are present in real world applications. A Bayesian network is a directed acyclic graph (DAG), in which nodes represent random variables and edges denote dependencies between them.[2]

There are three approaches for Bayesian learning when structure network is unknown: first approach is learning constraint-based, where Bayesian networks are seen as a representation of dependencies. In approach score-based, Bayesian networks are treated as a specification of a statistic model and then Bayesian learning is addressed to problem of model selection. In third approach instead to learn only one structure, it generate a set of feasible structures. This methods increase Bayesian reasoning, and try to average the prediction for each structure that belong to the set of possible structures.

The approach score-based attempt to find a network that optimizes a selected scoring function, which evaluates the fitness of each feasible network to the data. The scoring functions can be formulated based on different principles, such as, inter alia, Likelihood and Bayesian scores. The optimization procedures, that is, the task to finding a network structure that optimizes the scoring function is a combinatorial optimization problem, and is known to be NP-hard [1, 7]. Hense, the optimization process often stops at a local optimal structure.

Bayesian learning based in Markov Chain MonteCarlo (MCMC) typically works by simulating a Markov chain over the space of feasible networks structures, whose stationary distribution is the posterior distribution of the network. A non-exhastive list of the work in this category include [9], [10] and [5], where the simulation is done using the Metropolis-Hasting (MH) algorithm [11, 6],and the network features are inferred by averaging over a large number of network simulated from the posterior distribution. Averaging over different networks significantly can reduce the risk suffered by the single model-based inference procedure. Although

CONTACT Diego M. Garcia diego.mauricio.garcia@correounivalle.edu.co

the approach seem attractive, they can only work well for the problems with a very small number of variables. As known by many researchers, the MH algorithm is prone to get trapped into a local energy minimum indefinitely in simulations from a system for which the energy landscape is rugged. To alleviate this difficulty, [4] introduce a two-stage algorithm: use the MH algorithm to sample a temporal order of nodes, and the sample a network structure compatible with the given order. As discussed in [4], for any Bayesian network, there exits a temporal order of the nodes such that for any two nodes X and Y, if there is an edge from X and Y, then X must be preceding to Y in the order. The two-state algorithm does improve the mixeing over the space of network structures, however, the structures sampled by it does not follow the correct posterior distribution, because the temporal order does not induce a partition of the space of network structures. A network may be compatible with more than one order. Refer to [3] for more discussions on this issue.

Based on [7], we describe in this paper how to learn Bayesian network usign a combination of the approach scored-based and MCMC. We attempt through maximum likelihood principle to finding a network structure that optimizes the likelihood function by simulating a Markov chain over the space of feasible networks structures, where the simulation is done using the MH algorithm. In learning structure, however, we are also concerned about the performance of the learned network on new instances sampled from the same underlying distribution P^*. Unfortunately, in this respect, the likelihood score can run into problems.
The likelihood score is a good measure of the *fit* of the estimated Bayesian network and the training data, but the maximum likelihood score *never* prefers the simpler network over the more complex one and it assigns both networks the same score only in these rare situations when their variables are truly independent in the training data. Adding an edge to a network structure can never decrease the maximum likelihood score, therefore, the more complex network will have a higher score in all but a vanishingly small fraction of cases, despite that there are situations where we should prefer to learn the simpler network (for example, when variables are *nearly independent* in the training data).
When we use a data set D (training data) to define an *empirical distribution* ($\hat{P}_D$ is a probability distribution), the maximum likelihood network will exhibit a conditional independence only when that independence happens to hold exactly in the empirical distribution. Due to statistical noise, exact independence almost never occurs, and therefore, in almots all cases, the maximum likelihood network will be a *fully connected one*. In other words, the *likelihood score overfits the training data*, learning a model that precisely fits the specifics of the empirical distribution in our training set. this model therefore fails to generalize well to new data cases: these are sampled from underlying distribution, which is not identical to the empirical distribution in our training set.
Since the likelihood score does not provide us with tools to avoid overfitting, we have to be careful when using it. It is reasonable to use the maximum likelihood score when there are additional mechanisms that disallow overly complicated structures. To alleviate this difficulty, we proposes some strategies to avoid overfitting, for example, we will discuss learning networks with a fixed indegree. Such a limitation can constrain the tendency to overfit when using the maximum likjelihood score. As well, we propose before adding an edge validate whether variables implicated in new edge are *nearly independent* in the training data. Finally, we propose a strategy post-optimal as filter of complex structures, in which, we apply filter of near-independence to each edge of the set of optimal structures.

The remainder of this paper is organized as follow. In Section 2, we give the formulation of Bayesian networks. In Section 3, we first give a brief review of the MH algorithm and describe its implementation for Bayesian networks. In Section 4, we describe the forward sampling algorithm to obtain the training set D and we propose strategies to avoid overfitting of MCMC Bayesian learning. In Section 5, we present the numerical results on a simulated example and we conclude the paper with a brief discussion.

2 Bayesian networks

A Bayesian network model can be defined as a pair $B = (G, \rho)$. Here, $G = (\upsilon, \varepsilon)$ is a directed acyclic graph that represents the structure of the network, υ denotes the set of nodes, and ε denotes the set of edges. Each element of the parameter vector ρ represents a conditional probability.[2]

Example problem: Lung cancer. A patient has been suffering from shortness of breath (called dyspnoea) and visits the doctor, worried that he has lung cancer. The doctor knows that other diseases, such as

(a) Set of nodes and values for the lung cancer problem.

Node name	Values
Pollution	{*low, high*}
Smoker	{*T, F*}
Cancer	{*T, F*}
Dyspnoea	{*T, F*}
X-ray	{*pos, neg*}

(b) Set of edges for the lung cancer problem.

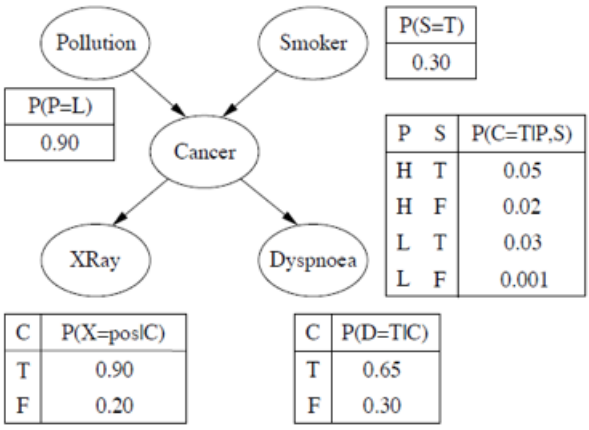

Figure 1: A BN for the lung cancer problem.[8]

tuberculosis and bronchitis, are possible causes, as well as lung cancer. She also knows that other relevant information includes whether or not the patient is a smoker (increasing the chances of cancer and bronchitis) and what sort of air pollution he has been exposed to. A positive X-ray would indicate either TB or lung cancer.

Representing the lung cancer problem across from a Bayesian network, the set of nodes υ could be as you can see in the figure **1a**. In this case (Discrete)

For a node $\nu_i \in \upsilon$, a parent of ν_i is a node from which there is a directed link to ν . The set of parents of ν_i is denoted by $pa(\nu_i)$. As you can see in the figure **1b**, if $\nu_i = Cancer$ then $pa(\nu_i) = [Pollution, Smoker]$.

3 Learning Bayesian networks using MCMC

3.1 Markov chains

3.2 Metropolis Hasting

4 Materials and methods

Sectioning commands work just as they do in the `article` document class.

4.1 Materials

If you followed how sectioning commands work, then you might have guessed how to get a subsection.

4.2 Methods

In fact, this document class was built using the `article` document class. Hopefully, if you started with that document class or something analogous to it, converting to this document class is not too difficult.

5 Results and Discussion

Getting the hang of it yet?

5.1 Without strategy

5.2 Ilustration of strategies

5.2.1 Strategy One

5.2.2 Strategy Two

5.2.3 Strategy Three

References

[1] David Maxwell Chickering. *Learning Bayesian Networks is NP-Complete*, pages 121–130. Springer New York, New York, NY, 1996.

[2] Dey, D.K. and Ghosh, S. and Mallick, B.K. *Bayesian Modeling in Bioinformatics*. Chapman & Hall/CRC Biostatistics Series. CRC Press, 2010.

[3] Byron Ellis and Wing Hung Wong. Learning causal bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482):778–789, 2008.

[4] Nir Friedman and Daphne Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 50(1-2):95–125, 2003.

[5] P Giudici and PJ Green. Decomposable graphical gaussian model determination. *Biometrika*, 86(4):785–801, 1999.

[6] W. K. HASTINGS. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[7] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[8] K.B. Korb and A.E. Nicholson. *Bayesian Artificial Intelligence, Second Edition*. Chapman & Hall/CRC Computer Science & Data Analysis. CRC Press, 2010.

[9] David Madigan and Adrian E Raftery. Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

[10] David Madigan, Jeremy York, and Denis Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.

[11] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.