Neurocomputing 150 (2015) 404–416

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Weighted ensemble learning of Bayesian network for gene regulatory networks

Hasna Njah *, Salma Jamoussi

MIRACL (Multimedia InfoRmation system and Advanced Computing Laboratory), University of Sfax, Tunis Street km. 10, Technopole, Sfax, Tunisia

ARTICLE INFO

Article history:
Received 31 December 2013
Received in revised form
18 May 2014
Accepted 21 May 2014
Available online 8 November 2014

Keywords:
Ensemble learning
Cluster analysis
Bayesian network
Bayesian network fusion
Microarray data

ABSTRACT

Gene Regulatory Network (GRN) is known as the most adequate representation of genes' interactions based on microarray datasets. One of the most performing modeling tools that enable the inference of these networks is a Bayesian network (BN). When preceded by an efficient pre-processing step, BN learning can unveil possible relationships between key disease genes and allows biologists to analyze these interactions and to exploit them. However, the layout of microarray data is different from classic data. This particularity engenders challenges to BN learning in terms of dimensionality and data over-fitting.

In this paper, we propose a fuzzy ensemble clustering method that allows outputting small and highly inter-correlated partitions of genes so that we can overcome dimensionality problem. We present a weighted committee based structure algorithm for learning BNs of each partition without over-fitting training dataset. Moreover, we offer an approach for assembling the sub-BNs through genes in common. We also statistically verify and biologically validate our approach.

1. Introduction

Although microarrays emerged recently, they have known a fast technological development. This development upgraded them from being unreliable and money-consuming to precise and more affordable. Microarray data extraction and mapping open wide opportunities for decoding the mysteries behind the behavior of genes.

In bio-informatics, there is a multitude of research areas and software that analyze genes' expressions and relationships. Gene Regulatory Networks (GRNs) were exposed as the most efficient and performing graphical and user-friendly representations of these relations. They are commonly represented as a graphical model where nodes refer to genes, or other elements of interest in the regulatory system, and edges refer, by their directed or undirected form, to interactions or associations between genes. GRNs partially allow unveiling the relations that cause the emergence of some diseases.

Modeling GRNs can be ensured by different methods such as *the Boolean Network* [36] and *the System of Differential Equations* [17]. The former represents genes as sets of *nodes* and *binary variables* that quantify the gene expression into active (1) or inactive (0) states. This method represents the relationships between genes as sets of *edges* and *functions* composed of simple Boolean operations. The latter is a dynamic model that explicitly describes gene expression variation in time as a function of other genes' expression and environmental factors.

This panoply of methods does not prevent Bayesian Networks (BNs) [27] from being distinguished as a performing and flexible GRN model. As a matter of fact, BN learning allows discovering the regulatory relations among genes in a probabilistic context while dealing with the missing data. BN uses a qualitative representation that consists of a directed acyclic graph (DAG) and a quantitative representation that consists of a set of conditional probability tables (CPT). Each node presents a gene whose expression value is described through a continuous or a discrete random variable. Edges represent causal regulatory relations between genes. The probability of each gene expression level is computed using the probability of its parents.

A BN's DAG and CPTs can be established by an expert in case of small networks. However, large and complex problems, such as GRNs, cannot be simply modeled by a biologist. In such cases, the correspondent BN model has to be learnt based on a certain set of observations. Therefore, two types of machine learning methods are applicable: *Structure Learning* (SL) methods which enable finding the best DAG that describes data and *Parameter Learning* methods which ensure fitting CPTs to the data and the chosen structure.

* Corresponding author.
E-mail address: njah.hasna@gmail.com (H. Njah).

http://dx.doi.org/10.1016/j.neucom.2014.05.078
0925-2312/

The main objective of SL algorithms is to find the graph that best describes data using a training dataset. Generally, it is possible to search for interdependencies between variables. In that case, *Constraint-based methods* are used. Another possibility is to explore the space of possible structures and choose the one with the best score. The used algorithms, in that case, are called *Score based algorithms*. There is another alternative which benefits from the advantages of the two approaches. This is referred to as *Hybrid methods*.

Parameter learning step in the process of inferring Bayesian network is carried out using training data obtained from the real-world problem under interest. It is processed by either a *statistical method* (i.e. Maximum likelihood [58]) or by a *Bayesian approach* (i.e. Maximum a posteriori estimation [37]).

The construction of a BN, in the context of GRN modeling, needs a microarray dataset as input. The structure of the latter is quite different from the classic datasets. Unlike classical data, microarrays appear with a high number of features (gene expressions), which are about tens of thousands, and a relatively low number of samples, which are about dozens. This idiosyncrasy of microarray datasets gave birth to a dilemma in BN learning. On the one hand, it is required to take into account the great number of genes, which is quite impossible since SL is an NP-complete problem [12]. On the other hand, it is substantial to avoid over-fitting due to the limited number of instances.

As the enumeration of all possible structures becomes impossible for databases with a high number of features, an intuitive solution is to divide the training dataset into relatively small sub-datasets according to features. This operation is referred to as Gene Clustering. It can be either *hard*, which means that each gene belongs to a unique cluster, or *soft* (i.e. fuzzy clustering) which means that a gene can belong to different clusters with a given percentage. Gene clustering is considered to be the most faithful technique that expedite BN learning while preserving all genes involved in the biological context. However, it has to be rigorous and less fault-tolerant. Indeed, no clustering algorithm leads to a perfect result. Each technique can lead to promising results with datasets and fail with others. Therefore, it is beneficial to take into consideration a committee of known clustering algorithms from different families (i.e. partitioning methods, hierarchical methods, and density-based methods) and to apply a consensus clustering in order to avoid falling into a mediocre partition.

Furthermore, the limited number of instances can be resolved by the application of BN ensemble SL. In fact, SL algorithms use eventually different methods and optimizations to find the “best structure”. Each class of methods has its own characteristics. For instance, it can be considered that constraint-based approaches are more likely to find the best structure since they give indication about the reliability of the learnt network [64]. Also, it can be assumed that score-based approaches are better for learning structures since they take into account the statistical aspect of BNs. Moreover, it is assumed that the hybrid methods are the most efficient since they benefit from the advantages of constraint-based methods and score-based methods [46]. Yet, it is useful to learn a BN by taking into account the benefits of each method of the most known SL algorithms.

2. Related work to BN learning adaptation

2.1. Features clustering

Ensemble clustering techniques are surveyed in Ghaemi et al. [28] where the authors give an extensive state of the art presentation of these methods. They also provide a complexity comparison of the existent techniques, the pros and cons and a possible categorization of these methods. Other surveys could be found in Alizadeh et al. [3] and Ghosh and Acharya [29]. The authors of Naldi et al. [49] propose various strategies to rank the partitions of a partition set and include or exclude them from composing the consensus partition. The rank is based on the internal clustering validation indexes [41] and on external validation indexes measuring the dissimilarity between couples of partitions.

2.2. BN ensemble learning

Ensemble learning is an effective technique that has been increasingly adopted to combine multiple learning algorithms to improve the overall performance [18]. It basically aims at escaping over-fitting the training data. There are different ensemble learning methods that can be applied to enhance the performances of a BN learning module. It is convenient to group them according to various perspectives.

The first one is based on data perturbation by *bootstrap sampling* [21]. In that way, sub-sets of training dataset are re-generated and the best structure is chosen. It is possible to distinguish *non-parametric bootstrap methods*, where sub-datasets are generated directly from the training dataset [19], and *parametric bootstrap*, where sub-datasets are generated from an initially learnt BN [26].

The second one concerns *genetic algorithms* [4]. They take into account partial structures as solutions to BN learning problems. Then they use crossover and mutation to combine these structures in a global one while optimizing a fitness function.

Bootstrapping methods and genetic algorithms are known for their effectiveness against over-fitting. Also they are simply programmed. However, they fail at converging to a final BN in many cases.

Moreover, it is possible to select random subsets from the training dataset, apply a SL method on each subset and combine the learnt networks into a global one. This operation is inspired from the random forests' principle. Indeed, it is possible to be inspired from this tree-based classification method in order to propose an efficient BN learning method [66].

Another perspective is found on set-based or committee-based SL. In that manner, different SL algorithms are applied on the same training dataset. The combination of the learnt structures is ensured by fixing a threshold on the corresponding vote of each edge [46]. The final BN can be a quasi-essential graph of all BNs [50] or majority graph of the found BNs [1].

2.3. BNs merging

Two possible techniques were presented. The first one uses *module BN learning* [62] and the second one uses *BNs fusion*. The former consists of learning a sub-BN by the use of each cluster. Then it considers this sub-network as a node in a global BN. The resultant graph is, therefore, composed of a global BN whose nodes are intermediate BNs [57]. The latter consists of learning the sub-BNs of each cluster. Then it applies a BN fusion method in order to construct the global BN of the whole database. There are three main categories of BN fusion methods: *DAG fusion methods*, *Dependency-based methods* and *Data-based methods*.

DAG fusion methods focus on merging two DAGs while ensuring that the resulting graph does not contain cycles [33,74,44,54]. They can refer to prior knowledge on eventual arcs by consulting experts. Usually, CPTs are partially – or cannot even be – updated when using these methods.

Dependency based methods ensure BN fusion by applying dependence tests [24,60,70]. They usually test the dependencies between nodes in different clusters [53]. After finding the final structure, CPTs can be updated directly, in case the whole dataset