Pattern Recognition 42 (2009) 3046 – 3056

Contents lists available at ScienceDirect

Pattern Recognition

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

Learning Bayesian network parameters under incomplete data with domain knowledge

Wenhui Liao[a,*], Qiang Ji[b]

[a]Thomson Reuters, Eagan, MN 55123, USA
[b]ECSE, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ARTICLE INFO

Article history:
Received 24 April 2008
Received in revised form 4 February 2009
Accepted 7 April 2009

Keywords:
Bayesian network parameter learning
Missing data
EM algorithm
Facial action unit (AU) recognition

ABSTRACT

Bayesian networks (BNs) have gained increasing attention in recent years. One key issue in Bayesian networks is parameter learning. When training data is incomplete or sparse or when multiple hidden nodes exist, learning parameters in Bayesian networks becomes extremely difficult. Under these circumstances, the learning algorithms are required to operate in a high-dimensional search space and they could easily get trapped among copious local maxima. This paper presents a learning algorithm to incorporate domain knowledge into the learning to regularize the otherwise ill-posed problem, to limit the search space, and to avoid local optima. Unlike the conventional approaches that typically exploit the quantitative domain knowledge such as prior probability distribution, our method systematically incorporates qualitative constraints on some of the parameters into the learning process. Specifically, the problem is formulated as a constrained optimization problem, where an objective function is defined as a combination of the likelihood function and penalty functions constructed from the qualitative domain knowledge. Then, a gradient-descent procedure is systematically integrated with the E-step and M-step of the EM algorithm, to estimate the parameters iteratively until it converges. The experiments with both synthetic data and real data for facial action recognition show our algorithm improves the accuracy of the learned BN parameters significantly over the conventional EM algorithm.

1. Introduction

In recent years, Bayesian networks (BNs) have been increasingly used in a wide range of applications including computer vision [1], bioinformatics [2], information retrieval [3], data fusion [4], decision support systems and others. A BN is a directed acyclic graph (DAG) that represents a joint probability distribution among a set of variables, where the nodes denote random variables and the links denote the conditional dependencies among variables. The advantages of BNs can be summarized as their semantic clarity and understandability by humans, the ease of acquisition and incorporation of prior knowledge, the possibility of causal interpretation of learned models, and the automatic handling of noisy and missing data [5].

In spite of these claims, people often face the problem of learning BNs from training data in order to apply BNs to real-world applications. Typically, there are two categories in learning BNs, one is to learn BN parameters when a BN structure is known, and another is to learn both BN structures and parameters. In this paper, we focus on BN parameter learning by assuming the BN structure is already known. If the training data is complete, learning BN parameters is not difficult, however, in real world, training data can be incomplete for various reasons. For example, in a BN modeling video surveillance, the training data may be incomplete because of security issue; in a BN modeling customer behaviors, the training data may be incomplete because of privacy issue. Sometimes, the training data may be complete but sparse, because some events rarely happen, or the data for these events are difficult to obtain.

In general, training data can be missed in three ways: *missing at random (MAR)*, *missing completely at random (MCAR)*, and *not missing at random* (NMAR). MAR means the probability of missing data on any variable is not related to its particular value, but could be related to other variables. MCAR means the missing value of a variable depends neither on the variable itself nor on the values of other variables in the BN. For example, some hidden (latent) nodes never have data. NMAR means data missing is not at random but depends on the values of the variables.

The majority of the current learning algorithms assume the MAR property holds for all the incomplete training samples since learning is easier for MAR than NMAR and MCAR. The classical approaches

* Corresponding author.
E-mail addresses: wenhui.liao@thomsonreuters.com (W. Liao), qji@ecse.rpi.edu (Q. Ji).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.04.006

include the Expectation Maximization (EM) algorithm [6] and Gibbs sampling [7]. Other methods are proposed to overcome the disadvantages of EM and Gibbs sampling. For example, methods are proposed to learn the parameters when data are not missing at random, such as the AI&M procedure [8], the RBE algorithm [9], and the maximum entropy method [10,11]; some methods are proposed to escape local maxima under the assumption of MAR, such as the information-bottleneck EM (IB-EM) algorithm [12], data perturbation method [13], etc.; other methods are proposed to speed up the learning procedure, such as generalized conjugate gradient algorithm [14], online updating rules [15], and others.

When data are missing completely at random, in other words, when several hidden nodes exist, those methods could fail, where the learned parameters may be quite different from the true parameters. In fact, since there are no data for hidden nodes, learning parameters becomes an ill-posed problem. Thus, prior data on domain knowledge are needed to regularize the learning problem. In most domains, at least some information, either from literature or from domain experts, is available about the model to be constructed. However, many forms of prior knowledge that an expert might have are difficult to be directly used by existing machine learning algorithms. Therefore, it is important to formalize the knowledge systematically and incorporate it into the learning. Such domain knowledge can help regularize the otherwise ill-posed learning problem, reduce the search space significantly, and help escape local maxima.

This motivates us to propose a Bayesian network learning algorithm for the case when multiple hidden nodes exist by systematically combining domain knowledge during learning. Instead of using quantitative domain knowledge, which is often hard to obtain, we propose to exploit qualitative domain knowledge. Qualitative domain knowledge impose approximated constraints on some parameters or on the relationships among some parameters. These kind of qualitative knowledge are often readily available. Specifically, two qualitative constraints are considered, the range of parameters, and the relative relationships between different parameters. Instead of using the likelihood function as the objective to maximize during learning, we define the objective function as a combination of the likelihood function and the penalty functions constructed from the domain knowledge. Then, a gradient-descent procedure is systematically integrated with the Expectation-step (E-step) and Maximization-step (M-step) of the EM algorithm, to estimate the parameters iteratively until it converges. The experiments show the proposed algorithm significantly improves the accuracy of the learned BN parameters over the conventional EM method.

2. Related work

During the past several years, many methods have been proposed to learn BN parameters when data are missing. Two standard learning algorithms are Gibbs sampling [7] and EM [6]. Gibbs sampling by Geman and Geman [7] is the basic tool of simulation and can be applied to virtually any graphical model whether the arcs are directed or not, and whether the variables are continuous or discrete [16]. It completes the samples by inferring the missing data from the available information and then learns from the completed database (imputation strategy). Unfortunately, Gibbs sampling method suffers from convergence problems arising from correlations between successive samples [10]. In addition, it is not effective when data are missing in complete random (e.g. the case of the hidden nodes).

The EM algorithm can be regarded as a deterministic version of Gibbs sampling used to search for the Maximum Likelihood (ML) or Maximum a Posteriori (MAP) estimate for model parameters [16,6]. However, when there are multiple hidden variables or a large amount of missing data, EM gets easily trapped in a local maximum. "With data missing massively and systematically, the likelihood function has a number of local maxima and straight maximum likelihood gives results with unsuitably extreme probabilities" [17]. In addition, EM algorithms are sensitive to the initial starting points. If the initial starting points are far away from the optimal solution, the learned parameters are not reliable.

Different methods are proposed to help avoid local maxima. Elidan and Friedman [12] propose an information-bottleneck EM (IB-EM) algorithm to learn the parameters of BNs with hidden nodes. It treats the learning problem as a tradeoff between two information-theoretic objectives, where the first one is to make the hidden nodes uninformative about the identity of specific instances, and the second one is to make the hidden variables informative about the observed attributes. However, although IB-EM has a better performance than the standard EM for some simple BNs, it is actually worse than EM for the complex hierarchical models as shown in [12]. To escape local maxima in learning, Elida et al. [13] propose a solution by perturbing training data. Two basic techniques are used to perturb the weights of the training data: (1) random reweighing, which randomly samples weight profiles on the training data, and (2) adversarial reweighing, which updates the weight profiles to explicitly punish the current hypothesis, with the intent of moving the search quickly to a nearby basin of attraction. Although it usually achieves better solutions than EM, it is still a heuristic method and not necessarily able to escape local maxima. And also, it is much slower than the standard EM algorithm.

The previous methods emphasize improving the machine learning techniques, instead of using domain knowledge to help learning. Since there are no data available for hidden nodes, it is important to incorporate any available information about these nodes into learning. The methods for constraining the parameters for a BN include Dirichlet priors, parameter sharing, and qualitative constraints. According to [18], there are several problems using Dirichlet priors. First, it is impossible to represent even the simple equality constraints on the parameters. Second, it is often beyond expert's capability to specify a full Dirichlet prior over the parameters of a Bayesian network. Parameter sharing, on the other hand, allows parameters of different models to share the same values, i.e., it allows to impose equality constraints. Parameter sharing methods, however, do not capture more complicated constraints among parameters such as inequality constraints among the parameters. In addition, both Dirichlet priors and parameter sharing methods are restricted to sharing parameters at the level of sharing a whole CPT or CPTs, instead of at the level of granularity of individual parameters. To overcome these limitations, others [19–22,18] propose to explicitly exploit qualitative relationships among parameters and systematically incorporate them into the parameter estimation process.

Druzdel et al. [19] give formal definitions of several types of qualitative relationships that can hold between nodes in a BN to help specify CPTs of BNs, including probability intervals, qualitative influences, and qualitative synergies. They express these available information in a canonical form consisting of (in)equalities expressing constraints on the hyperspace of possible joint probability distributions, and then use this canonical form to derive upper and lower bounds on any probability of interest. However, the upper and lower bounds cannot give sufficient insight into how likely a value from the interval is to be the actual probability.

Wittig and Jameson [20] present a method for integrating formal statements of qualitative constraints into two learning algorithms, APN [23,24] and EM. Two types of qualitative influences [19] are considered as constraints for parameters during learning in this method: (1) a positive influence holds between two variables (X_1, X_2) if for any given value of X_2, an increase in the value of X_1 will not decrease the probability that the value of X_2 is equal to or greater than that given value; and (2) a negative influence can be defined analogously.