Expert Systems with Applications 36 (2009) 3497–3504

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Augmenting learning function to Bayesian network inferences with maximum likelihood parameters

WeiYi Liu, Kun Yue *, JiaDong Zhang

Department of Computer Science and Engineering, School of Information Science and Engineering, Yunnan University, Kunming 650091, PR China

ARTICLE INFO

Keywords:
Bayesian network
Inference
Learning function
Support vector machine
Maximum likelihood hypothesis

ABSTRACT

Computing the posterior probability distribution for a set of query variables by search result is an important task of inferences with a Bayesian network. Starting from real applications, it is also necessary to make inferences when the evidence is not contained in training data. In this paper, we are to augment the learning function to Bayesian network inferences, and extend the classical "*search*"-based inferences to "*search + learning*"-based inferences. Based on the support vector machine, we use a class of hyperplanes to construct the hypothesis space. Then we use the method of solving an optimal hyperplane to find a maximum likelihood hypothesis for the value not contained in training data. Further, we give a convergent Gibbs sampling algorithm for approximate probabilistic inference with the presence of maximum likelihood parameters. Preliminary experiments show the feasibility of our proposed methods.

1. Introduction

As a graphical representation of probabilistic causal relationships, Bayesian networks (BNs) are effective and widely used frameworks (Cooper & Herskovits, 1992; Heckerman & Wellman, 1995; Pearl, 1998; Russel & Norving, 2002). A Bayesian network can be constructed by means of statistical learning from sample data (Buntine, 1996; Cheng, Greiner, Kelly, Bell, & Liu, 2002; Pearl, 1987).

The basic task for probabilistic inference in a Bayesian network is to compute the posterior probability distribution for a set of query variables by search result, given some observed evidence. For example, in Fig. 1, we can deduce the probability distribution for *cholesterol standards* of somebody whose *age* is 60. However, in real applications, some queries are also addressed frequently given arbitrary evidence values. For example. if we know John is 65 years old, how to deduce the probability distribution for his cholesterol standards? The ordinary inference with the Bayesian network (in Fig. 1) by search cannot answer this question, since there is no data about patients of 65 years old in the training sample. This leads to our idea of extending "*search*"-based inferences to "*search + learning*"-based inferences with a Bayesian network. "*Learning*" means determining a hypothesis space H and finding the most probable hypothesis h in H, given the training sample.

Fortunately, support vector machine (SVM) is a new machine learning method based on the statistical learning theory. The support vector machine not only has solved certain problems in many learning methods, such as small sample, over fitting, high dimension and local minimum, but also has a fairly high generalization (forecasting) ability (Burges, 1998; Chang & Liu, 2001). These characteristics of SVMs make possible extend the general Bayesian network inferences by augmenting the learning function and obtain the desired hypothesis.

In this paper, our purpose is to discuss Bayesian network inferences with the learning function, as well as the network construction. We extend general Bayesian networks by augmenting maximum likelihood parameters to make the inference done when evidence values are not contained in training sample data. Based on the support vector machine, we use a class of hyperplanes to construct the hypothesis space and use the method of solving an optimal hyperplane to find a maximum likelihood hypothesis, in which both the linear and nonlinear cases are discussed. Thus we can obtain conditional probability tables of a Bayesian network including maximum likelihood parameters. This approach is not only extending the expressive power of a Bayesian network, but also finding a new application for SVMs.

Further, in this paper we give a Gibbs sampling algorithm for approximate probabilistic inference with the presence of maximum likelihood parameters.

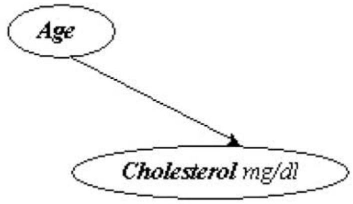
Preliminary experiments were conducted to test the accuracy of our proposed method for learning the maximum likelihood parameters and the convergence of the algorithm for corresponding approximate inferences. Experimental results show that our proposed methods are feasible.

The remainder of this paper is organized as follows. In Section 2, related work is introduced. In Section 3, we propose the method for

* Corresponding author.
E-mail addresses: liuweiyi2000@yahoo.com.cn (W. Liu), yue.kun@gmail.com (K. Yue).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.02.018

A	$P(A)$
a_1=60	0.3
a_2=70	0.2
a_3=80	0.3
a_4=90	0.2



$P(C\|A)$	a_1=60	a_2=70	a_3=80	a_4=90
c_1=150	2/3	0	0	0
c_2=170	1/3	1	1/3	0
c_3=190	0	0	2/3	1

Fig. 1. A simple Bayesian network about *Age* and *Cholesterol*.

maximum likelihood hypothesis learning based on SVM. In Section 4, we give the approximate inference algorithm of Bayesian networks with the learning function. In Section 5, experimental results are shown. At last, we conclude and discuss the future work in Section 6.

2. Related work

Bayesian networks have become an established framework for representing and reasoning with uncertain knowledge (Cooper & Herskovits, 1992; Heckerman & Wellman, 1995; Pearl, 1998; Russel & Norving, 2002), and have been used in many different aspects of intelligent applications (Liao, Wang, Li, & Liu, 2006; Heckerman, Mandani, & Wellman, 1995).

The construction of a Bayesian network involves structure learning and parameter learning. A variety of algorithms has been proposed to induct structures of Bayesian networks. These algorithms mainly fall into 2 categories: scoring&search based and dependency-analysis based algorithms. In the case where the network structure is given and the variable values are fully observed in the training sample data, learning the conditional probability table is straightforward. Nevertheless, if there are only some of the variable values are observable in the sample data, the learning problem is difficult. Russel and Norving (2002) proposed a similar gradient ascent procedure that learns the entries in conditional probability tables. However, complete probability tables are doomed to be constructed by Russel's algorithm.

Exact inference in large and connected networks is difficult (Pearl, 1998; Russel & Norving, 2002). Consequently, approximate inference methods are considered frequently, such as Monte Carlo algorithm (Russel & Norving, 2002). Based on conditional independence, Pearl defined the Markov blanket, which describes the direct causes and direct effects given a certain node in a Bayesian network (Pearl, 1998). The discovery of Markov blanket is applied in the Monte Carlo algorithm for Bayesian network inference.

As a novel statistic learning method, support vector machines have been paid wide attention recently (Burges, 1998; Chang & Liu, 2001). SVMs are based on the structural risk minimization principle from statistical learning theory (Vapnik, 1982; Vapnik, 1998). The idea of structural risk minimization is to find a hypothesis that can guarantee the lowest error. SVM classification is to construct an optimal hyperplane, with the maximal marginal of separation between 2 classes (Burges, 1998; Chang & Liu, 2001). Furthermore, by introducing the kernel function, SVMs can handle non-linear feature spaces, and carry out the training by considering combinations of more than one feature (Boser, Guyon, & Vapnik, 1992; Poggio, 1975; Schölkopf, Burges, & Smola, 1999). SVMs have been used for the biological sequence classification (Brown, Grundy, Liu, & Cristianini, 2000), text classification (Basu et al., 2003), patter recognition (Burges, 1998; Osuna et al., 1997), regression estimation (Smola, Bcholköpf, & Miiller, 1998), and so on. However, according to our knowledge, SVMs have not yet been applied to the Bayesian network inferences. In this paper, we are right to construct the Bayesian network with parameters of maximum likelihood hypothesis based on SVMs, as well as the corresponding approximate inference method.

3. Learning the maximum likelihood hypothesis based on SVMs

As mentioned in Section 1, we cannot make inference with a general Bayesian network when the given evidence is not contained in sample data, which is also frequently addressed in real applications. To conquer such difficulties, it is necessary to extend the Bayesian network by incrementally augmenting parameters of maximum likelihood hypothesis for those values that are not contained in sample data, instead of reconstructing the conditional probability tables completely. Thus, the question is how we can implement such extension to Bayesian networks and corresponding inferences. In this section, we first discuss the SVMs-based method for learning the maximum likelihood hypothesis as the desired parameters in a Bayesian network.

3.1. Maximum likelihood hypothesis

We first introduce the concept of maximum likelihood hypothesis (Mitchell, 1997).

Let us assume the training data D is of the form $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where y_i is the target function of the sample x_i. Note that each $x_i = \{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(m)}\}$ is a vector, where the component $x_i^{(l)}$ is called a feature, and y_i can have several values $y_i^{(1)}, \ldots, y_i^{(j)}, \ldots$, where $y_i^{(j)} \in \{-1, +1\}$. Let H be a hypothesis space, and each h in H is a function of the form $h : D \to R^m$. We write $P(h|D)$ to denote the posterior probability of h given the training data D. The most probable hypothesis $h \in H$ given the data D is called a *maximum a posterior* (MAP) hypothesis, written h_{MAP}

$$h_{\text{MAP}} = \arg\max_{h \in H} P(h|D).$$

By the Bayesian theorem, we have

$$h_{\text{MAP}} = \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} = \arg\max_{h \in H} P(D|h)P(h).$$

We assume that $P(h_i) = P(h_j)$ for all h_i and h_j in H, and thus we need only consider the term $P(D|h)$.

$P(D|h)$ is called the *likelihood* of data D given h, and any hypothesis that maximizes $P(D|h)$ is called a *maximum likelihood hypothesis* h^*

$$h^* = \arg\max_{h \in H} P(D|h).$$

When x_i is independent of h, we can write $P(D^{(k)}|h)$ as

$$P(D^{(k)}|h) = \prod_{i=1}^{n} P(x_i, y_i^{(k)}|h) = \prod_{i=1}^{n} P(y_i^{(k)}|h, x_i) \cdot P(x_i),$$

where $D^{(k)} = \{(x_1, y_1^{(k)}), (x_2, y_2^{(k)}), \ldots, (x_n, y_n^{(k)})\}$.$P(y_i^{(k)}|h, x_i)$ is the probability of $y_i^{(k)} = 1$ for an instance x_i given h. Note that h is the hypothesis regarding the target function, which computes the above conditional probability, and thus $P(y_i^{(k)} = 1|h, x_i) = h_{y_i^{(k)}}(x_i)$.

From this point of view, the inference with a Bayesian network can be described as follows: let $y_1^{(k)}$ and x_i be the target and evi-