Knowledge-Based Systems 22 (2009) 489–495

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

On the classification performance of TAN and general Bayesian networks

Michael G. Madden *

College of Engineering and Informatics, National University of Ireland, University Road, Galway, Ireland

ARTICLE INFO

Article history:
Available online 9 January 2009

Keywords:
Bayesian networks
TAN
Naïve Bayes
Classification
Inductive learning
Parameter estimation

ABSTRACT

Over a decade ago, Friedman et al. introduced the Tree Augmented Naïve Bayes (TAN) classifier, with experiments indicating that it significantly outperformed Naïve Bayes (NB) in terms of classification accuracy, whereas general Bayesian network (GBN) classifiers performed no better than NB. This paper challenges those claims, using a careful experimental analysis to show that GBN classifiers significantly outperform NB on datasets analyzed, and are comparable to TAN performance. It is found that the poor performance reported by Friedman et al. are not attributable to the GBN per se, but rather to their use of simple empirical frequencies to estimate GBN parameters, whereas basic parameter smoothing (used in their TAN analyses but not their GBN analyses) improves GBN performance significantly. It is concluded that, while GBN classifiers may have some limitations, they deserve greater attention, particularly in domains where insight into classification decisions, as well as good accuracy, is required.

1. Introduction

This paper examines the performance of Bayesian networks as classifiers, comparing their performance to that of the Naïve Bayes (NB) classifier and the Tree Augmented Naïve Bayes (TAN) classifier, both of which make strong assumptions about interactions between domain variables.

In the experiments performed for this work, described below in Section 3, standard Bayesian networks (referred to as General Bayesian Networks, GBNs, to distinguish them from NB and TAN) are compared with NB and TAN classifiers on 28 standard benchmark datasets. Our experiments indicate that the GBN classifier is substantially better than NB, with performance closer to that of TAN. This contrasts with the conclusions drawn in the landmark paper on Bayesian network classifiers by Friedman et al. [14]. That paper presented results on many of the same datasets, showing that GBNs constructed using the minimum description length (MDL) score tend to perform no better than NB. That result has been widely noted by other authors (e.g. [16,18]); in one case the result was interpreted as indicating that NB "easily outperforms" GBN.

Our contention is that it has become 'accepted wisdom' that GBN classification performance is no better than that of NB, and significantly worse than TAN (ignoring other considerations such as computational complexity or interpretability). Our results indicate that GBN's classification performance is superior to that of NB and much closer to that of TAN, when the same parameter estimation procedure is used for all.

It turns out that Friedman et al. used simple frequency counts for parameter estimation in constructing GBN classifiers, whereas they used parameter smoothing in constructing TAN classifiers (see Section 2.3 for details). Our experiments show that if frequency counts are used for both GBN and TAN, neither is much better than NB (Section 3.3, Fig. 5), but if parameter smoothing is used for both, they both perform similarly well (Fig. 4). Furthermore, since GBN classifiers are commonly constructed through heuristic search, it is possible for improved GBN construction algorithms to lead to improved performance.

The structure of the paper is as follows. Section 2 reviews Bayesian networks and the algorithms for constructing GBN and TAN classifiers that are used in this paper. Section 3 presents experiments applying NB, TAN and two GBN algorithms to classification problems on 28 standard datasets, and identifies why the results of this paper are at odds with those of Friedman et al. as mentioned above. Finally, Section 4 draws general conclusions about the suitability of GBNs as classifiers.

2. Bayesian networks and classification

As is well known, a Bayesian network is composed of the network structure and its conditional probabilities. The structure B_S is a directed acyclic graph where the nodes correspond to domain variables $x_1,\ldots,x_n$ and the arcs between nodes represent direct dependencies between the variables. Likewise, the absence of an arc between two nodes x_1 and x_2 represents that x_2 is independent of x_1 given its parents in B_S. Using the notation of Cooper and Herskovits [12], the set of parents of a node x_i in B_S is denoted π_i. The structure is annotated with a set of conditional probabilities, B_P,

* Tel.: +35391493797; fax: +35391444214.
E-mail address: michael.madden@nuigalway.ie.

0950-7051/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.10.006

containing a term $P(X_i|\Pi_i)$ for each possible value X_i of x_i and each possible instantiation Π_i of π_i.

2.1. Inductive learning of Bayesian networks

Several algorithms have been proposed since the late 1980s for inductive learning of general Bayesian networks. Recent developments include the global optimization approach of Silander and Myllymäki [23], the Greedy Equivalence Search algorithm [9], and the Three-Phase Dependency Analysis algorithm [8], though this latter algorithm has subsequently been shown to be incorrect [10]. We evaluate two approaches to GBN construction, described in the following sub-sections, both of which approaches have relatively low computational complexity:

1. The K2 search procedure [12] in conjunction with the Bayesian BDeu scoring metric [5], which is a refinement of the K2 metric.
2. The approach used by Friedman et al. [14], which combines hill-climbing search with the MDL score.

These are both search-and-score methods for construction of GBNs; a search heuristic is used to propose candidate networks, and a scoring function is used to assess, for any two candidates, which one is more likely given the training data.

The scoring functions and search procedures are described in greater detail in the following sub-sections. Rather can constructing general BN structures, restrictions may be placed on the structures; this is described in Section 2.2. Typically, the conditional probabilities (parameters) associated with a network are not computed from the data until after the structure has been found; parameter estimation is described in Section 2.3.

2.1.1. K2: search with BDeu scoring approach

If D is a database of training cases, Z is the set of variables in each case in D, and B_{Si} and B_{Sj} are two belief network structures containing exactly those variables that are in Z, then the comparison amounts to calculating $P(B_{Si}|D)/P(B_{Sj}|D)$, which in turn reduces to calculating $P(B_{Si},D)/P(B_{Sj},D)$.

Assume that Z is a set of n discrete variables, where a variable x_i in Z has r_i possible value assignments, $(v_{i1},\ldots,v_{iri})$, and that D has N cases, each with a value assignment for each variable in Z. A network structure B_S is assumed to contain just the variables in Z. Each variable x_i in B_S has zero or more parents, represented as a list π_i. Let w_{ij} denote the jth unique instantiation of π_i relative to D, and assume that there are q_i such unique instantiations of π_i. Let N_{ijk} be defined as the number of cases in D in which variable x_i has the value v_{ik} and π_i is instantiated as w_{ij}. Let N'_{ijk} denote a Dirichlet parameter. Let N_{ij} and N'_{ij} be defined as:

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}, \quad N'_{ij} \equiv \sum_{k=1}^{r_i} N'_{ijk} \tag{1}$$

With these definitions, the BD metric [17] is defined as:

$$P(B_S, D) = P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \tag{2}$$

Note that Γ is the gamma function, defined as $\Gamma(x+1) = x\Gamma(x)$, which is closely related to the factorial function but defined for real numbers, not just integers. In a practical implementation, the logs of terms in Eq. (2) are computed.

The K2 metric [12] corresponds to Eq. (2) with all Dirichlet exponents set to 'uninformative' values of $N'_{ijk} = 1$. Alternative uninformative values are proposed by Buntine [5]:

$$N'_{ijk} = \frac{N'}{r_i q_i} \tag{3}$$

Using Buntine's values, Eq. (2) becomes what Heckerman et al. [17] term the BDeu metric, which has the additional property of being structure-equivalent. This is the metric used in the current work. Assuming that all structures are equally likely *a priori*, $P(B_S)$ is constant, so to maximize $P(B_S,D)$ just requires finding the set of parents for each node that maximizes the second inner product of Eq. (2).

The K2 search procedure requires a node ordering. It operates by initially assuming that a node has no parents, and then adding incrementally that parent whose addition most increases the probability of the resulting network. Parents are added greedily to a node until the addition of no one parent can increase the structure probability. This is repeated for all nodes in the sequence specified by the node ordering.

In the experiments of Section 3, the node ordering in each dataset is arbitrarily taken to be the order of attributes in the input files, except that the class node is always placed first in the order. In addition, the maximum number of parents a node may have is limited to 4.

2.1.2. MDL scoring approach

In constructing GBNs, Friedman et al. [14] use a scoring function based on the minimum description length (MDL) principle. The MDL score of a network B given a database of training cases D is:

$$\mathrm{MDL}(B|D) = \frac{1}{2}\log N|B| - LL(B|D) \tag{4}$$

where $|B|$ is the number of parameters in the network and $LL(B \mid D)$ denotes the log-likelihood of B given D. To calculate $LL(B \mid D)$, let $\hat{P}_D(\cdot)$ be the empirical probability measure defined by frequencies of events in D. Then:

$$LL(B|D) = N \sum_i \sum_{X_i,\Pi_i} \hat{P}_D(X_i, \Pi_i) \log(\hat{P}_D(X_i|\Pi_i)) \tag{5}$$

The search procedure used by Friedman et al. is to start with the empty network and successively apply local operations that greedily reduce the MDL score maximally until a local minimum is found. The local operations applied are arc insertion, arc deletion and arc reversal.

2.1.3. Classification using a GBN

A Bayesian network may be used for classification as follows. Firstly, any nodes outside of the Markov blanket of the classification node x_c may be deleted. Then, assume that the value of x_c is unknown and the values of all other nodes are known. Then, for every possible instantiation of x_c, calculate the joint probability of that instantiation of all variables in the network given the database D. By the definition of a Bayesian network, the joint probability of a particular instantiation of all n variables is calculated as:

$$P(x_1 = X_1, \ldots, x_n = X_n) = \prod_{i=1}^{n} P(x_i = X_i|\pi_i = \Pi_i) \tag{6}$$

By normalizing the resulting set of joint probabilities of all possible instantiations of x_c, an estimate of the relative probability of each is found. The vector of class probabilities may be multiplied by a misclassification cost matrix, if available. Note that the classification node is not considered 'special' when building the GBN, and in Eq. (6), x_c is just one of the variables $x_1,\ldots,x_n$.

Although arbitrary inference in a GBN with discrete variables is NP-hard [11], the classification procedure just described just requires Eq. (6) to be evaluated once for each possible instantiation of x_c; thus its time complexity is $O(n_m r_c)$, where n_m is the number of nodes in x_c's Markov blanket; $n_m \leqslant n$.