Explaining Bayesian network classifiers

Wassihun belay ID: GSE/0461/15

AI masters EXTENTION Student

1. Abstract

In this paper, we empirically evaluate algorithms for learning four types of Bayesian network (BN) classifiers - Nai've-Bayes, tree augmented Nai've-Bayes, BN augmented Nai've-Bayes and general BNs, . Comparing it based on measure the learning accuracy and classification efficiency. computational time. these Experimental results argue that which BN classifiers deserve more competitive with (or superior to) the best known classifiers, based on both Bayesian networks and other formalisms; methodology use to experimental results were carried out using eight datasets downloaded from the UCI machine learning repository

2. Introduction

The domain of this paper is explaining about Bayesian network classification Because a Bayesian network is a complete model for its variables and their relationships, it can be used to answer probabilistic queries about them. It is powerful tools for knowledge representation and inference under conditions of uncertainty, and then classify an observation on features based on the posterior marginal of the class variable. Crucially, Bayesian networks can also be used to predict the joint probability over multiple outputs (discrete and or continuous). for predicting a discrete class variable C. It assigns x, an observation of n predictor variables (features) X = (X1, . . . , Xn), to the most probable class .Bayesian networks are different from other knowledge-based systems tools because uncertainty is handled in mathematically rigorous yet efficient and simple way. Many of the Bayesian network construction algorithms are based on the “node sequence already known” condition and the purpose is to reduce and simplify the complexity of the structure.

3. Objective

The general objective of this paper is explaining Bayesian networks as classifiers as examines the performance of, comparing their performance then use experimental results.

3.1 Specific Objectives

In general, a BN can be used to compute the conditional probability of one node, given values assigned to the other nodes; hence, a BN can be used as a classifier that gives the posterior probability distribution of the classification node given the values of other attributes. When learning Bayesian networks from datasets, we use nodes to represent dataset attributes. In SECTION ONE I try to explain types of Bayesian network classifier and in section two compare this types of BN classifiers based on real world dataset to make experimental results on which BN classifier is better for which dataset.

4. Discussion(frame work)

4.1 Naive-Bayes

A Naive-Bayes BN, as discussed in (Duda and Hart, 1973), is a simple structure that has the classification node as the parent node of all other nodes (see Figure 1). No other connections are allowed in a Naive-Bayes structure.

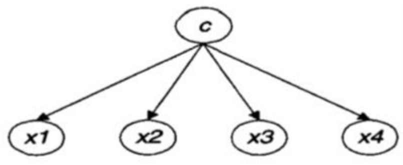

Figure 1: A simple Naïve Bayes structure

It has two advantages over many other classifiers due to its assumption that all the features are independent of each other. First, it is easy to construct, as the structure is given a priori (and hence no structure learning procedure is required). Second, the classification process is very efficient. Although this independence assumption is obviously problematic, Naive-Bayes has surprisingly outperformed many sophisticated classifiers over a large number of datasets, especially where the features are not strongly correlated (Langley et al. 1992).

4.2 TREE AUGMENTED NAİVE-BAYES (TAN}

Let X= {x1,….x_n , c} represent the node set (where c is the classification node) of the data. T h e algorithm for learning TAN classifiers (Friedman et al. 1997) first learns a tree structure over X \{c} , using mutual information tests conditioned on c. It then adds a link from the classification node to each feature node, similar to a Naive-Bayes structure (i.e., the classification node is a parent of all other nodes) - see Figure 2. (Note that features x_1,x_2,x_3,x_4 form a tree.)

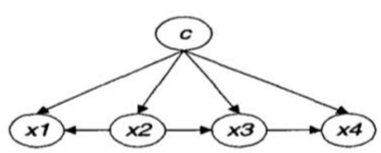

Figure 2: A simple TAN structure

This complete algorithm, which extends the Chow-Liu algorithm, requires O(N^2) conditional mutual information tests.

4.3 BN AUGMENTED NAIVE-BAYES (BAN)

BAN classifiers extend TAN classifiers by allowing the attributes to form an arbitrary graph, rather than just a tree (Friedman et al. 1997) -- see Figure 3. The BAN-learning algorithm is just like the TAN learner of Section 4.2, but the Step 2 of the BAN-learner calls an unrestricted BN-I earning algorithm instead of a tree learning algorithm.

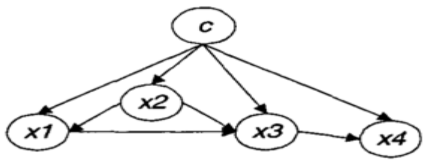

Figure 3: A simple BAN structure

Like the TAN-learning algorithm, this algorithm does not require additional mutual information tests, and so it requires $O(N^2)$ mutual information tests.

4.4 GENERAL BAYESIAN NETWORK (GBN)

Unlike the other BN-classifier learners, the GBN learner treats the classification nodes as an ordinary node (see Figure 4). The learning procedure is described as follows.

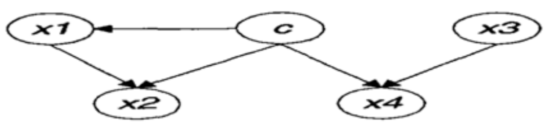

Figure 4: A simple GBN

This algorithm also information tests. requires $O(N^2)$ mutual

5 . EXPERIMENTS (METHODOLOGY)

The experiments were carried out using eight datasets downloaded from the UCI machine learning repository. The datasets it used are summarized in Table I. (CV5 stands for five-fold cross validation.)

Table 1: Datasets used in the experiments.

Dataset	Attributes.	Classes	Instances	
			Train	Test
Adult	13	2	32561	16281
Nursery	8	5	8640	4320
Mushroom	22	2	5416	2708
Chess	36	2	2130	1066
DNA	60	3	2000	1186
Car	6	4	1728	CV5
Flare	10	3	1066	CV5
Vote	16	2	435	CV5

The experiments were carried out as follows. first used the four learning algorithms presented in Section 4 to learn the four classifiers (one of each type) and then export the BNs to Bayesian Interchange Format (BIF v0.15) files. The GBN and BAN classifiers were learned using the default threshold setting of the Power Constructor. (This threshold determines how much mutual information between two nodes is considered as significant- see Section 4.3.) TAN and Naive-Bayes learning algorithms have no such threshold.

To test these classifiers on the test sets, use a modified version of JavaBayes (Cozman 1998). added some classes to the JavaBayes v0.34l so that it can read the test datasets and perform classification given a BN. The classification of each case in the test set is done by choosing, as class label, the value of class variable that has the highest posterior probability, given the instantiations of the feature nodes. The classification accuracy is measured by the percentage of correct predictions on the test sets (i.e., using a 0-1 loss function).

6 . RESULTS

Table 2 provides the prediction accuracy and standard deviation of each classifier.

From Table 2 we can see that the GBN, BAN and TAN have better overall performance than the Naive-Bayes in these experiments. BAN did best on four of the datasets and GBN and TAN each did best on two of the datasets.

Table 2: Running time (CPU seconds) of the classifier learning procedure.

	GBN	BAN	TAN
Adult	515	536	131
Nursery	10	11	10
Mushroom	136	134	38
Chess	41	56	37
DNA	300	570	113
Car	1	1	1
Flare	3	3	2
Vote	8	8	3

In experiments, it found that the classification process is also very efficient. JavaBayes can perform 100 to 600 classifications per second depending on the complexity of the classifier.and also found that the GBNs are often faster than Naive-Bayes on classification when the GBNs contain only a subset of the features.

7. EXTENSIONS

As we mentioned earlier, the GBN and BAN classifiers were learned using the default threshold setting given by Power Constructor. Based on the experience, this threshold setting is appropriate for most domains when the dataset is large enough. When the dataset is small, however, too high a setting will cause missing edges, and too low a setting will cause overfitting. Both situations will decrease the prediction accuracy of GBN and BAN. From Table 2, we can see that GBN or BAN can produce outstanding results when the datasets are large enough for their domains.

Comparing Bayesian Network Classifiers 107 Table 3: Experimental Results

	GBN (No. Selected FeaJTotal No. Fea.)	BAN	TAN	Naive-Bayes	Best reported
Adult	86.11±0.27 (8113)	85.82±0.27	86.01±0.27	84.18±0.29	85.95
Nursery	89.72±0.46 (6/8)	93.08±0.39	91.71:t0.42	90.32±0.45	N/A
Mushroom	99.30±0.16 (5/22)	100	99.82±0.08	95.79±0.39	100
Chess	94.65±0.69 (19/36)	94.18±0.72	92.50±0.81	87.34±1.02	99.53±0.21
DNA	79.09:t1.18 (43/60)	88.28±0.93	93.59±0.71	94.27±0.68	96.12±0.6
Car	86. II:t1.46 (5/6)	94.04±0.44	94.10±0.48	86.58±1.78	N/A
Flare	82.27±1.45 (1-3/10)	82.85±2.00.	83.49±1.29	80.11±3.14	83.40±1.67
Vote	95.17±1.89 (10-11116)	95.63±3.85	94.25±3.63	89.89±5.29	96.3:tl .3

8. CONCLUSION

In this paper, it empirically evaluated and compared four BN classifiers. The unrestricted classifiers (GBN and BAN), learned using two variants of the CBLl algorithm, give very encouraging results without using any wrapper function. After analyzing the experimental results, it proposed a wrapper algorithm that wraps around GBN and BAN, then demonstrated that this wrapper classifier can give even better results than the GBN and BAN classifiers.

Table 4 Experimental results using the wrapper

	GBN		BAN		Wrapper	Best reported
	Fixed Optimal thresh«>ld threshold		Fixed threshold	Optimal threshold	(best ofthe GBN and BAN using optimal thresholds)	
Adult	86.11±0.27	unchanged	85.82±0.27	unchanged	86.11±0.27	85.95
Nursery	89.72±0.46	90.32±0.45	93.08±0.39	95.74±0.31	95.74±0.31	NIA
Mushroom	99.30±0.16	unchanged	100	unchanged	100	100
Chess	94.65±0.69	94.09±0.72	94.18±0.72	96.44±0.57	96.44±0.57	99.53±0.21
DNA	79.09± 1.18	95.95±0.57	88.28±0.93	94.69±0.65	95.95±0.57	96.12±0.6

From the experiments it can also see that the time expenses of unrestricted BN-learning are only (at most) a few times slower than that of the efficient TAN learning. This is due to the three-phase learning mechanism used in the CBL l algorithm.

Given the theoretical analysis (Friedman et al. 1997) and the empirical comparison (results using scoring based methods on some of the data sets it use are reported in Friedman et al. 1997; Singh and Provan 1996), it believe that methods based on CI tests (such as mutual information tests) are more suitable for BN classifier learning than the more-standard scoring based methods. Note, in addition, that such mutual information tests are standard in decision tree learning and feature selection.

Another advantage of these learners is that they are constraint based. This means we can incorporate domain knowledge relatively easily, by just adding them as additional constraints. Such additional information will lead to yet better classification accuracy. The experiments also show that the classification can be performed very efficiently using even-general BN classifiers. Based on these results i believe that an improved type of BN classifiers, such as the ones shown here, should be used more often in real-world data mining and machine learning applications.