Learning Goal Oriented Bayesian Networks for Telecommunications Risk Management

Kazuo J. Ezawa
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
kje@ulysses.att.com

Moninder Singh
University of Pennsylvania
Dept. of Computer & Information Science
Philadelphia, PA 19104-6389
msingh@gradient.cis.upenn.edu

Steven W. Norton
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
norton@ulysses.att.com

Abstract

This paper discusses issues related to Bayesian network model learning for unbalanced binary classification tasks. In general, the primary focus of current research on Bayesian network learning systems (*e.g.,* K2 and its variants) is on the creation of the Bayesian network structure that fits the database best. It turns out that when applied with a specific purpose in mind, such as classification, the performance of these network models may be very poor. We demonstrate that Bayesian network models should be created to meet the specific goal or purpose intended for the model.

We first present a goal-oriented algorithm for constructing Bayesian networks for predicting uncollectibles in telecommunications risk-management datasets. Second, we argue and demonstrate that current Bayesian network learning methods may fail to perform satisfactorily in real life applications since they do not learn models tailored to a specific goal or purpose. Third, we discuss the performance of "goal oriented" K2 and its variant.

1 INTRODUCTION

Most current research on Bayesian network based learning systems (e.g., K2 [Cooper and Herskovits 1992], CB [Singh and Valtorta 1995], as well as the methods of [Heckerman, Geiger and Chickering 1994, Lam and Bacchus 1993, Bouckaert 1993]) focuses on the creation of Bayesian network structure that fits the database. These models are learned to "best" fit the data and do not take into account the specific purpose of the creation of the models *i.e.,* classification tasks and classification accuracy of the resultant models. Note that even though a given model may be "correct" in the sense of representing the structural relationships of the data, it need not be the best model when it comes to using it for a specific goal of making predictions [Cowell *et. al*. 1993]. We show that such methods may not work well in real life applications where the models should be learned for a specific goal (*e.g.* classification). First, we present a goal-oriented algorithm for constructing Bayesian networks using datasets in telecommunications risk-management. Second, we compare the performance of a goal oriented Bayesian network model (based on the Advanced Pattern Recognition & Identification system APRI) with that of a general Bayesian network model (based on K2) using telecommunications risk-management datasets. Lastly, we present a modified, goal oriented version of K2 and compare its performance to that of APRI.

Every year, the telecommunications industry incurs several billion dollars in uncollectible debt. Hence, controlling uncollectibles is an important problem in the industry. One of the key elements of risk management is the ability to use large quantities of historical data to build models for risk assessment on a per customer or per transaction basis.

If we can identify, with high accuracy, customers who will not pay their bills, or phone accounts for which we cannot collect, risk management would be simple. Instead, we can never really be certain about a customer or an account. That is not to say, though, that we must be entirely uninformed. To support risk-management policies that reduce the level of uncollectible debt, we need only provide an estimate of

the probability of uncollectible debt. In fact, unqualified black and white assessments will not be particularly useful because of the uncertainties that nonetheless come into play. Instead, a probability model could and should be devised as input to a normative decision-support system [Ezawa 1993]. That way a variety of reasoned actions might be considered, ranging from inaction to call disconnect. The question to be addressed is how to feasibly develop a useful probability model.

The datasets employed here contain customer-summary information of 40-90 thousand records and 20-50 million bytes which can be considered very large in machine learning research, but tiny in the telecommunications industry. The interesting outcomes are the non-paying customers, comprising just a few percent of the population. Unequal misclassification costs compound the difficulties. Non-paying customers that initially slip through undetected will be identified within a couple of billing cycles anyway. As bad as that may be, the greater potential problem is incorrectly classifying valuable paying customers. In today's highly competitive telecommunications market, dissatisfied customers have a range of options to choose from; the corresponding revenue might well be lost forever. The data are described by more than 45 variables, some discrete and some continuous. Many of the discrete variables have large unordered outcome sets. The continuous variables are not normally distributed. And last but not least, missing values are all too common.

Some learning methods simply cannot hope to process this much data or more in a timely manner because they process it many times over before converging to a solution [Baldi and Chauvin 1991]. Efficient decision tree learners that use recursive partitioning [Quinlan 1993] often have difficulty with discrete variables such as countries or city names and their large unordered outcome sets. In addition, their pruning mechanisms are easily thwarted by widely disparate class proportions. Under these conditions they often return a single node tied to the majority class rather than a meaningful tree structure. Even though we enriched our datasets by selecting a subpopulation more likely to be uncollectible, now 9 to 12% uncollectible instead of just a few percent uncollectible, these systems still have trouble characterizing the minority class. Lastly, state-of-the-art inductive learners offer little support for problem domains with unequal misclassification costs. To our knowledge, none has considered misclassification costs that vary from example to example the way they do in this domain. At the moment, appropriate treatment of unequal misclassification costs is an open research area [Catlett 1995, Pazzani *et al* 1994]. All of this is merely to illustrate the kinds of difficulties this data poses to learning systems in general, whether they are regression systems, nearest-neighbor systems, neural networks, *etc*.

One of the systems described in this paper is the Advanced Pattern Recognition and Identification (APRI) system, a Bayesian supervised machine-learning system. Comparisons between APRI and standard methods such as discriminant analysis and recursive tree builders can be found elsewhere [Ezawa and Schuermann 1995a,b]. Comparison of the performance of several conditionally-independent probabilistic models to the performance of conditionally-dependent models constructed by APRI using large call-detail datasets of 4-6 million records and 600-800 million bytes can be found in [Ezawa and Norton 1995].

2 THE BAYESIAN NETWORK APPROACH

Theoretically, the *Bayesian Classifier* [Fukunaga 1990] provides optimal classification performance. As a practical matter, however, its implementation is infeasible. Recent advances in evidence propagation algorithms [Shachter 1990, Lauritzen and Spiegelhalter 1988, Pearl 1988, Jensen *et al* 1990, Ezawa 1994] and computer hardware allow us to *approximate* the ideal Bayesian classifier by using Bayesian network models [Cheeseman 1988, Cooper and Herskovits 1992, Buntine and Smyth 1993, Langley and Sage 1994, Singh and Valtorta 1995]. This section describes Bayesian networks in general, the APRI learning algorithm, and a number of alternative approaches for learning Bayesian network models.

2.1 THE "GOAL ORIENTED" BAYESIAN NETWORK

The classification problem can be addressed using the joint probability $P(\pi, \mathbf{X})$ of classes or populations π and the variables $\mathbf{X}$ that describe the data.[1] In particular, an observation $\mathbf{X}$ can be classified as an instance of class

[1] Bold-faced capital letters will be used to denote vectors of features or attributes, such as entire observations. Individual features or attributes will be identified by subscripted capital letters. Particular values of an individual feature will be identified by further subscripting the corresponding lower-case letters.

π if π is most probable according to the conditional probability distribution $P(\pi \mid \mathbf{X})$. Assessing $P(\pi \mid \mathbf{X})$ directly is often infeasible due to data and storage limitations. The conditional probability of the attributes given the classes, $P(\mathbf{X} \mid \pi)$, and the unconditional probability of the classes, $P(\pi)$, are often assessed instead by analyzing a preclassified training data set. With those probabilities in hand, Bayes' rule then yields the desired conditional probability $P(\pi \mid \mathbf{X})$.

Merely representing $P(\pi, \mathbf{X})$ can be difficult with a large number of variables, if the distribution does not have a convenient structure. *Bayesian Networks* can be used to encode a wide variety of probabilistic information about probability distributions by factoring them into attribute level relationships. In a Bayesian network, a variable's parents can be thought of as its causes. Hence the factorization of $P(\pi, \mathbf{X})$ is easy to provide. For the network depicted in Figure 1, that factorization is as follows:

$$P(\pi, \mathbf{X}) = P(\pi) \times P(X_1 \mid \pi) \times P(X_2 \mid \pi)$$
$$\times P(X_3 \mid X_2, \pi) \times \ldots$$

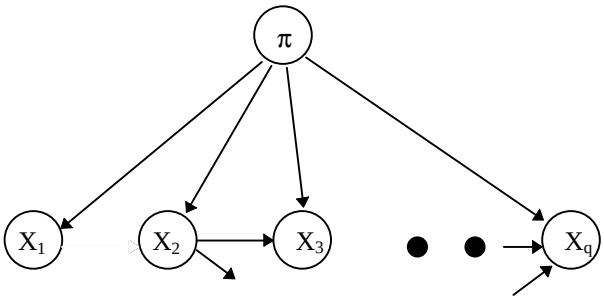

Figure 1: A Bayesian Network Model

The variables in this particular "Goal Oriented" Bayesian network have a common parent, namely the class node π. This is not true of all Bayesian networks. Instead, it is a feature of networks learned by APRI that helps them address the classification problem at hand. Besides the convenient decomposition of $P(\pi, \mathbf{X})$, Bayesian network models have a number of useful cognitive and computational properties that are described elsewhere [Pearl 1988]. One of the advantages that Bayesian networks provide is a graphical representation of the independence in a dataset. In Figure 1, for example, variable X_1 becomes independent of the other variables, once the true classification π is known.

2.2 APRI

The Advanced Pattern Recognition & Identification (APRI) system developed at AT&T Bell Laboratories is a Bayesian network-based supervised machine learning system that constructs graphical probability models like the one just described, using the entropy-based concept of mutual information to perform dependency selection [Cover and Thomas 1991]. It uses mutual information thresholds, first to select a set of variables and then to select a set of dependencies among the chosen variables. We settled on this approach to reduce training time with special emphasis on repeated reading of the training data. APRI is extremely efficient in this regard, never reading the training data more than five times.

APRI constructs graphical probability models in a four-step process. Three inputs are required: a database of training cases and two parameters, $T_{\pi x}$ and T_{xx}, each ranging between zero and one. $T_{\pi x}$ governs variable selection (or equivalently, selection of links between the class node and the variable nodes). T_{xx} governs selection of variable-to-variable links.

In the first step, APRI parses the input database and characterizes its variables. If the class variable is continuous, APRI first defines the class outcomes either by discretization or kernel density estimation. APRI then scans the database to identify the outcome sets for each variable. For continuous variables it either estimates the kernel density or uses information-based discretization.

In the second step, APRI chooses the variables for the final model. It computes the mutual information between the class node and the individual variables, then ranks the variables accordingly. Mutual information is related to the statistical concept of entropy:

$$H(X) = -\sum_i P(X_i) \log P(X_i)$$

The entropy of a variable can be thought of as a measure of the *a priori* uncertainty over its outcome. Mutual information is a symmetric function that can be defined directly in terms of probabilities:

$$I(X_I; X_J) = \sum_i \sum_j P(x_{Ii}, x_{Jj}) \log \frac{P(x_{Ii}, x_{Jj})}{P(x_{Ii})\, P(x_{Jj})}$$

or in terms of entropy and conditional entropy:

$$I(X_I; X_J) = H(X_I) - H(X_I \mid X_J)$$

The high-level idea is that the mutual information between X_I and X_J measures the reduction in uncertainty of X_I due to knowledge of X_J. It is zero if X_I and X_J are statistically independent. Its largest value, $H(X_I)$, is attained if X_I is a function of X_J. In a classification problem, a variable satisfying $I(X; \pi) = I(\pi; X) = H(\pi)$ would be very useful indeed. In fact, in that case the value of X uniquely determines the value of π.

Without loss of generality, let the indices from 1 to K provide the mutual-information ranking of the initial variables, so that $I(\pi; X_1) \geq I(\pi; X_2) \geq I(\pi; X_3)$ and so on. APRI selects the smallest number of variables J out of the entire pool of K variables, such that:

$$\sum_{j=1}^{J} I(\pi; X_j) \geq T_{\pi x} \sum_{k=1}^{K} I(\pi; X_k)$$

In other words, the parameter $T_{\pi x}$ establishes a mutual information threshold for choosing relevant variables. A value of 1 indicates that all the variables should be incorporated in the model. Lesser values indicate that the least informative variables should be excluded. In APRI's final graphical model, the class node becomes the parent of each of the selected variables.

The third step is akin to the second one, save that it identifies relationships between variables. In particular, it computes the conditional mutual information $I(X_i ; X_j \mid \pi)$ between pairs of the J previously identified variables, where $i \neq j$. These candidate links are rank ordered. The highest ranked are then selected until the cumulative value is just T_{xx} times the total conditional mutual information. Directionality of these links is based on the mutual information variable ranking determined in the second step, with higher ranked variables pointing towards lower ranked ones.

In the fourth and final step, APRI estimates $P(\pi)$ and $P(X_i \mid C(X_i))$ using frequency counts, where $C(X_i)$ represents the parents or causes of X_i, including the class node π.

2.3 ALTERNATIVE METHODS

A number of other authors have developed algorithms that search for graphical probability models by computing joint probabilities $P(B_s, D)$, where B_s is a Bayesian network structure and D a dataset [Cooper and Herskovits 1992, Heckerman *et al* 1994, Singh and Valtorta 1995]. These algorithms use evaluation functions that rank individual elements in the space of all Bayesian network structures. Their greedy search strategies use the probability metric to evaluate the networks that result from every possible incremental change to the current network, then apply the best of the changes and iterate. K2 is one such program [Cooper and Herskovits 1992]. This element-by-element approach was not adopted for our projects because it is impractical for use with massive datasets.

Programs like K2 face a potential, possibly serious, run-time problem. While K2 (and its variants) poses no problems for domains where the size of the database is small enough for the entire data to be read into memory, the number of times the algorithm reads the data may become a critical factor (as far as the efficiency of the process is concerned) for databases that are very large and have to be read from secondary storage, whenever required. If a dataset is too large to hold in memory, it must be read at least once for each arc in the final graphical model. For K2, it appears that the dataset would have to be read $O(n(u+1))$ times to create a model, where *n* is a number of nodes and *u* the maximum number of parents per node. With 33 variables and allowing just 2 parents per node, K2 might need to read the dataset 99 times. If the training data consists of several million records and perhaps hundreds of millions of bytes of data, as in the typical application in the telecommunication's risk-management, reading and re-reading the data becomes the limiting factor. APRI is very efficient in this regard, reading the database just four or five times during model creation, for any *n* and *u* : once in the first step for discrete classes or twice for continuous classes, then once in each of the three remaining steps.

We verified the performance of K2 by implementing a modified version of K2 under APRI system environment using account summary datasets from two different periods.

2.3.1 The K2 Algorithm:

We implemented K2 to replace APRI's second and third steps of feature selections of the model creation. Our implementation of K2 uses the first step of APRI for input data processing, and use the fourth (final) step to compute probabilities. The algorithm puts all the necessary information in memory, and uses APRI's variable rank order (*i.e.*, mutual information ranking) to set the order of variables for the model creation. We also experimented with CB [Singh and Valtorta 1995] for the second and third steps. Use of sub-modules of APRI was needed, since K2 does not provide facilities

to handle continuous variables, nor the ability to classify test datasets.

The K2 algorithm attempts to select the network which maximizes the network's posterior probability, $P(B_s, D)$. However, since it is computationally infeasible to search for the most probable network by exhaustively enumerating all possible Belief network structures (the number of possible networks exponential in the number of network nodes), the algorithm reduces the search space by requiring a total ordering on the features from which the network will be constructed. Then, given an ordering n_1, n_2, ..., n_m of the *m* attributes, the algorithm allows a node n_i to have parents only from the set of nodes $n_1,...,n_{i-1}$ that precede it in the ordering. The algorithm takes each successive attribute in the ordering, adds it as a node n_i in the network, and creates parents for n_i in a greedy fashion: rather than evaluate all subsets of network nodes $n_1,...,n_{i-1}$ as parent nodes, K2 selects as a parent node the single node in $n_1,...,n_{i-1}$ which most increases the posterior probability of the resultant network structure. New parent nodes are added incrementally to n_i as long as doing so increases the posterior probability of the network given the data.

The CB algorithm [Singh and Valtorta 1995] uses conditional independence tests to generate a "good" node ordering from the data, and then uses the K2 algorithm to generate the Bayesian network from the database using this node ordering. In particular, CB starts with the complete, undirected graph on all variables. It first deletes edges between adjacent nodes that are unconditionally independent (conditional independence tests of order 0). When the edges in the resultant graph are oriented, a total ordering on the variables is obtained. This ordering is then used in the K2 algorithm to construct the corresponding network. The algorithm then repeats this process by removing edges (from the undirected graph obtained in the previous iteration) between adjacent nodes that are conditionally independent given one node (conditional independence test of order 1). It keeps constructing networks for increasing orders of conditional independence tests as long as the predictive accuracy of the resultant network increases.

2.3.2 Goal Oriented K2:

We implemented two goal oriented versions of K2 described below. First, the original K2 algorithm was used with the additional constraint that each attribute have the class node, "uncollectible", as a parent. The intuition behind this was that each attribute should have a direct effect on the class variable in order to make a significant impact on classification accuracy. However, since forcing all variables in the model to have the class variable as a parent may introduce spurious dependencies (a variable may be conditionally independent of the class variable given some of the other variables), we also modified this approach to first select a subset of the variables that provide the maximum information about the class variable (based on conditional mutual information [Singh and Provan 1995]), and then use K2 to construct a Bayesian network from the selected set of variables with the constraint that each one of these variables should have the class node as a parent. The key difference between APRI and this Goal Oriented K2 is that the former selects all field nodes in one step and all field-to-field dependencies in one step, whereas the latter will select one node or one arc between field nodes at a time.

K2 as well as other current Bayesian network learning algorithms try to find the model that fits the data best, and do not care about the predictive accuracy of the resultant model. If, however, the Bayesian network is to be used for prediction, then techniques which learn Bayesian networks with this specific goal in mind might be expected to do better. Cowell and his colleagues [Cowell, Dawid and Spiegelhalter 1993, Spiegelhalter, Dawid, Lauritzen and Cowell 1993] have proposed the use of "global metrics" for measuring the quality of a Bayesian network model in terms of its classification accuracy. Algorithms for learning Bayesian networks based on such metrics should perform better than K2 and other related algorithms when models are to be learned specifically for classification. We have not tested "global or local" metrics.

3 COMPARISON OF BAYESIAN NETWORK MODELS USING TELECOMMUNICATIONS RISK-MANAGEMENT DATASETS.

In this section we first compare APRI, K2, and CB, and then compare APRI and the Goal Oriented K2 and CB. The APRI model was constructed using a 95% attribute selection threshold and 25% attribute-to-attribute threshold ($T_{\pi x}$ and T_{xx} respectively).

The training set includes 68,138 collectible and 6,633 uncollectible accounts described by 21 attributes. The unconditional probability of being uncollectible is 9.7%. Our final task is to predict/classify a dataset from a subsequent period using a model created from the first period dataset, since in the end, when this

model is applied to real world business scenarios, it will be trained on one period and asked to perform in another later period. The testing sample sizes are 94,004 collectibles and 10,481 uncollectibles, yielding an 11.1% unconditional probability of being uncollectible.

One of the interesting features of predicting uncollectible debt is the requirement of genuine out-of-sample testing datasets from separate time periods. Such testing is essential because of the inevitable lag between model creation and model deployment. Of course, there is the risk that fraud or uncollectible patterns will change in the interim. Seasonal variations could even interfere. Given enough data, these effects might be modeled. In addition, active network policies could also change observed patterns of activity, although there is probably less hope of modeling the effects of untried policies. Despite these potential pitfalls, subsequent-period out-of-sample prediction will remain the real litmus test for this application.

One of the conventional strategies for comparing probability distributions and scoring models is ROC analysis [Swets 1988]. ROC analysis (a shorthand for Receiver Operating Characteristic analysis) grew out of WWII research on signal detection as a means to help understand the trade-off between false positives and true positives. In our application, the true positives are the uncollectible accounts correctly classified as uncollectible, while the false positives are collectible accounts incorrectly classified as uncollectible. Given a classifier and a set of testing data, the numbers of true positives and false positives can be computed directly. But a probability model is not a classifier, and so the number of true positives and the number of false positives will vary according to the model and the probability threshold that distinguishes positive instances from negative instances. We used "predicted" uncollectible-probability thresholds from 0.0 to 0.9 to generate the ROC curves given in Figure 2. One ROC curve is preferred to another if it is above or to the left. That is, the more desirable curve has a lower false positive rate for a particular true positive rate.

3.1 APRI VS K2

Figure 2 shows the performance of APRI, conditionally independent or "naive" Bayes, K2, and CB models. APRI's performance is far superior to that of K2 and CB. We also found K2 to be very sensitive to the ordering of variables provided for the model creation (K2 does not provide the ordering of variables/nodes). The K2 model presented here uses the ordering of nodes provided by APRI (mutual information ranking in step 2).

In our experiments K2 was a poor performer in every instance when compared to APRI with its goal oriented model. Because K2 aims to model a probability distribution and not to solve a classification problem *per se*, it constructed models that did not bear, for the most part, on the class node *uncollectible*. K2 and CB models even performed worse than the conditionally independent model.

Figure 3 depicts the K2 model structure derived by using APRI's variable ordering. Variables X_i named after from the order of importance indicated by APRI, *i.e.*, π, X_1, X_2, .., and X_{21}. K2 created an elaborate Bayesian network incorporating various variable relationships that generally made intuitive sense. On the other hand, the classification node π has only the one successor X_2. Hence, the prediction of π is solely dependent on X_2 unless the value of X_2 is unknown. Unfortunately in this dataset there were no missing values of X_2. Hence the rest of the variables had no impact on the classification π.

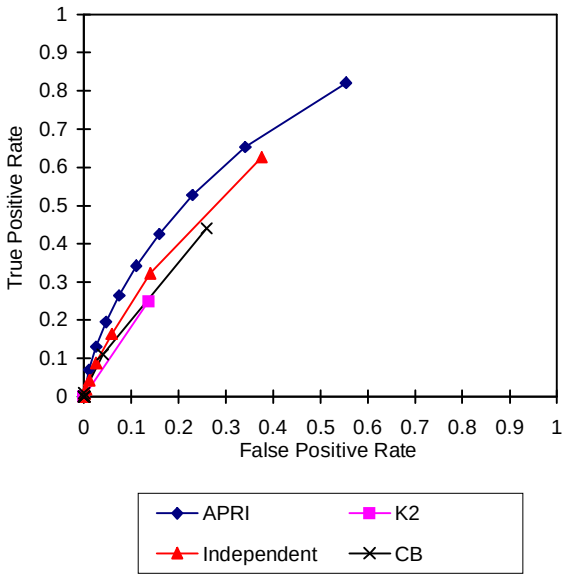

Figure 2: APRI vs Modified K2 ROC Curves

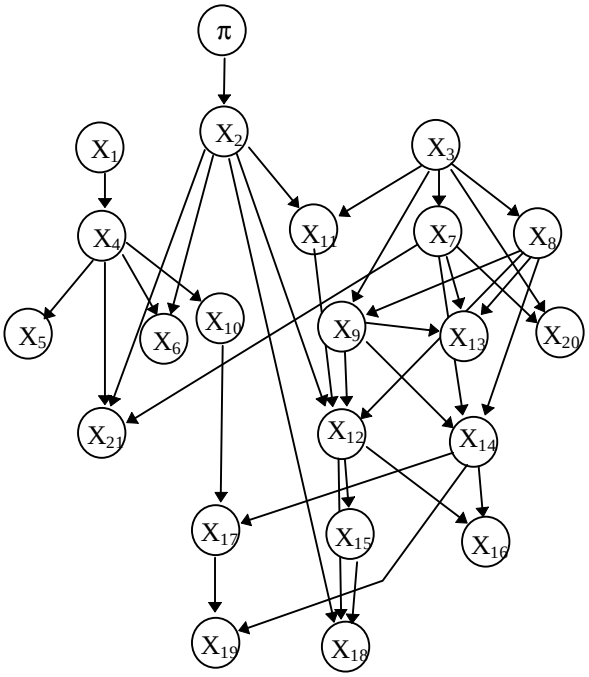

Figure 3: K2

Figure 4 shows the APRI model structure. All the variables retained in the model have the classification node π as a predecessor, *i.e.*, only variables which impact the classification of π are retained. Hence it is not surprising that the APRI model out-performs K2 and its variant CB.

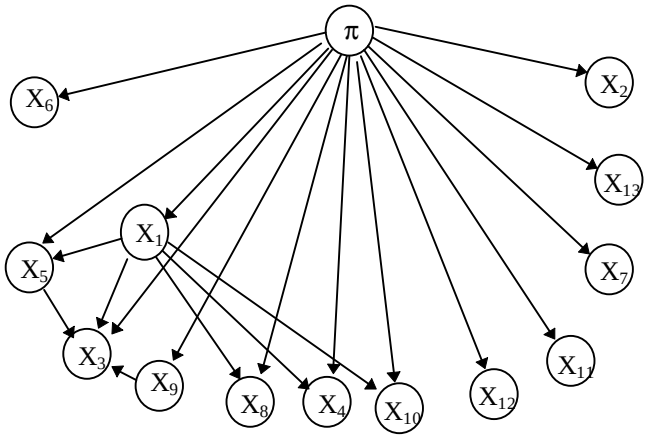

Figure 4: APRI model

In terms of predicted probability of uncollectible for K2 model, none of the test data records were classified as uncollectible with probability of 20% or higher. Since background probability of uncollectible was 11.1%, it is difficult to use this probability to justify our action against any account.

In the machine learning datasets, it has been shown that in general K2 and its variant, CB work competitively with other learning methods such as C4.5 [Singh and Provan, 1995]. In the domain as difficult as ours, it turns out that blindly applying general Bayesian network models without any specific goal is a futile exercise. This lead to the further modification of K2 with specific goal (class/prime node).

3.2 APRI vs Goal Oriented K2

Figure 5 shows ROC curve analysis of APRI, conditionally independent (naive Bayes), Goal Oriented K2, and Goal Oriented CB. APRI performs slightly better than the rest of the models. The performance of the modified K2, modified CB, and the naive (independent) models seem equivalent. Note that for both K2 and CB models, all the variables retained in the model have π as parents. The difference among APRI, K2, and CB models are that they have different dependencies among the variable retained in the model. K2 and CB show signs of overfitting. Clearly in terms of classification accuracy, their model expansion criteria of increased posterior probability of the network given the data doesn't seem be appropriate here.

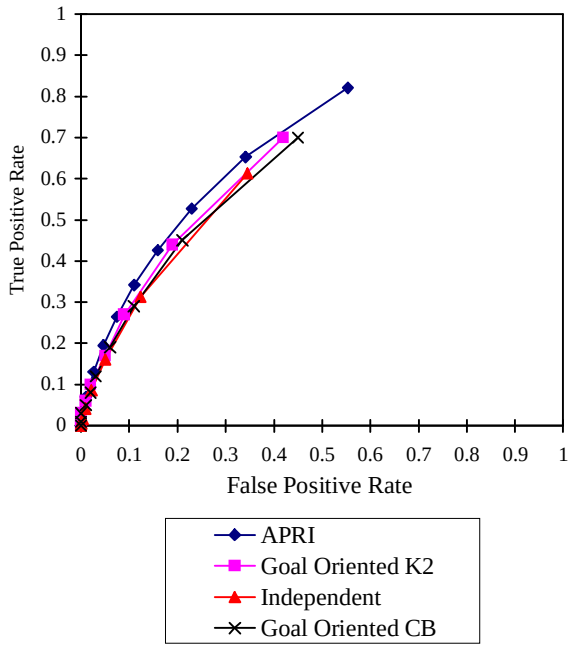

Figure 5: APRI vs Goal Oriented K2s ROC Curves

On the other hand, the results are not surprising. In any real-world application, all the variables are carefully analyzed statistically and screened out (*e.g.*, irrelevant or duplicating variables) before the variables appear on the dataset. It goes through a few filters before it appears in the final dataset. Hence, whether an algorithm selects a variable one at a time or selects several at once doesn't seem to produce significant differences. With a specific goal for the creation of the model, the performance from different methods for the creation of Bayesian network seem to converge.

4 DISCUSSION AND SUMMARY

In this paper, we discussed issues related to Bayesian network models in unbalanced binary classification performance. In general, most of the current research on Bayesian network based learning systems (*e.g.*, K2, and its variants) focus on the creation of the Bayesian network structure that fits the database best. It turns out that when applied to a specific purpose such as classification, the performance of these network models may be very poor.

We presented a goal-oriented algorithm for constructing Bayesian networks for predicting uncollectibles in telecommunications risk-management datasets. We discussed and demonstrated that the general Bayesian network learning like K2 is not necessarily suited for a specific purpose of classification task using telecommunication's risk-management datasets.

We discussed and proposed goal oriented K2 and its variant CB, and compared the performance with APRI. The classification performance of goal oriented K2 and CB improved dramatically from those of general K2 and CB. We demonstrate that the Bayesian network model should be created to meet the specific goal/purpose of the model in mind.

References

Baldi, P., and Chauvin, Y., (1991). Temporal Evolution of Generalization During Learning in Linear Networks. *Neural Computation*, **3**, pp. 589-603.

Bouckaert, R.R., (1993). Belief Network Construction using the Minimum Description Length Principle. *Proceedings ECSQARU*, pp. 41-48.

Buntine, W. L. and Smyth, P., (1993). Learning from Data: A Probabilistic Framework. Tutorial Program, Ninth Conference on Uncertainty in Artificial Intelligence.

Catlett, J., (1995). Tailoring Rulesets to Misclassification Costs. Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statistics, pp. 88-94.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D., (1988). AUTOCLASS: A Bayesian Classification System. *Proceedings of the Fifth International Conference on Machine Learning*, pp. 54-64, Morgan Kaufmann.

Cooper, G. F. and Herskovits, E., (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, **9**, pp. 309-347.

Cover, T. M. and Thomas, J. A., 1991, *Elements of Information Theory*, John Wiley and Sons.

Cowell, R. G., Dawid, A. P., Spiegelhalter, D. J., (1993). Sequential Model Criticism in Probabilistic Expert Systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 15, No. 3, pp. 209-219.

Ezawa, K. J., (1993). A Normative Decision Support System. *Proceedings of the Third International Conference on Artificial Intelligence in Economics and Management*.

Ezawa, K. J., (1994). Value of Evidence on Influence Diagrams. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 212-220, Morgan Kaufmann.

Ezawa, K. J., and Norton S., (1995) Knowledge Discovery in Telecommunication Services Data Using Bayesian Networks. *Proceedings of the First International Conference on Knowledge Discovery & Data Mining*, Montreal, Canada.

Ezawa, K. J., and Schuermann, T., (1995a). Fraud/Uncollectible Debt Detection Using a Bayesian Network Based Learning System: A Rare Binary Outcome with Mixed Data Structures. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 157-166, Morgan Kaufmann.

Ezawa, K. J. and Schuermann, T., (1995b). A Bayesian Network Based Learning System: Architecture and Performance Comparison with Other Methods. In C. Froideveaux, J. Kohlas (Eds.), *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Lecture Notes in Artificial Intelligence **946**, 197-206, Springer, Berlin.

Fukunaga, K., (1990). *Introduction to Statistical Pattern Recognition*, Academic Press.

Heckerman, D. E., Geiger, D., and Chickering D. M., (1994). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, pp. 293-301.

Jensen, V., Olesen K. G., and Anderson S. K., (1990). An Algebra of Bayesian Universes for Knowledge-Based Systems. *Networks*, **20**, pp. 637-659.

Lam, W. and Bacchus, F., (1994). Learning Bayesian Belief Networks, an approach based on the MDL principle. *Computational Intelligence*, **10**, No. 4.

Langley, P. and Sage, S., (1994). Induction of Selective Bayesian Classifiers. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 399-406, Morgan Kaufmann.

Lauritzen, S. L., and Spiegelhalter, D. J., (1988). Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. *J. R. Statistics Society, B*, **50**, No. 2, pp. 157-224.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., and Brunk, C., (1994). Reducing Misclassification Costs. *Proceedings of the International Conference on Machine Learning*, pp. 217-225, Morgan Kaufmann

Pearl, J., (1988). *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann.

Quinlan, J. R., (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann.

Shachter, R. D., (1990). "Evidence Absorption and Propagation through Evidence Reversals", *Uncertainty in Artificial Intelligence*, Vol. 5, pp. 173-190, North-Holland.

Singh, M. and Provan, G., (1995). Efficient Learning of Selective Bayesian Network Classifiers. Technical Report No. MS-CIS-95-36, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA 19104.

Singh, M. and Valtorta, M., (1995). Construction of Bayesian Network Structures from Data: a Brief Survey and an Efficient Algorithm. *International Journal of Approximate Reasoning*, **12**, pp. 111-121.

Spiegelhalter, D.J., Dawid, A. P., Lauritzen, S.L., and Cowell, R.G., (1993). Bayesian Analysis in Expert Systems. *Statistical Science*, Vol. 8, No. 3, 219-283.

Swets, J. A., (1988). Measuring the Accuracy of Diagnostic Systems. *Science*, Vol. 240, pp. 1285-1293.