Construction of Bayesian Classifiers with GA for Predicting Customer Retention*

Hongmei Shao
College of Math. and Comput. Sci.
China University of Petroleum
Dongying, 257061, China

Gaofeng Zheng
JANA Solutions, Inc.
Shiba 1-15-13, Minato-ku
Tokyo, 105-0014, Japan

Fengxian An
Dept. of Comput. Sci.
Huaiyin Inst. of Tech.
Huaian, 223001, China

Abstract

In this paper, a novel Bayesian classifier model is constructed based on genetic algorithms (GA) for the prediction of customer churn. Experiment results have shown that it not only matches potential churning customers well from the large amount of validation data but also shows higher classifying precision compared with the other three Bayesian classifier models. Recently, the model has been adopted by a Japanese enterprise to successfully handle some business problems of potential churning customers prediction.

1. Introduction

With the coming of economy globalization and information times, the concept of people's consumption has greatly changed in the past two decades. An typical example is the popularity of credit cards in the world, especially in many western developed countries. To boom their business and gain more profit, many credit card suppliers have changed their traditional management policy to 'customers-center' and focused on Customer Relationship Management (CRM). As an important business scenario within CRM, customer retention has been perceived as a central topic in their business management and marketing decisions [1].

The prior discussion of the tangible advantages of retaining customers appeared in 1990 by Dawkins and Reichheld [2]. In the following years, other scholars pointed out that customer retention management has the potential for delivering substantial benefits to firms in terms of long-term profitability, based on the analysis of the strong association between customer retention and profitability: long-term customers buy more and are less costly to serve, whereas replacing existing customer by 'new' ones is known to be more expensive [3–7]. Thus, how to retain current loyal customers and reduce the potential churning customers has become more and more important.

It is impossible, however, to be absolutely certain that a customer will still be loyal or not because there is no way to know a person's intent. Given the reality, the best we can do is to construct a probability model to estimate the happening probability of the above two cases for him by using historical data about transaction and analyzing the cause of past churning. With the rising of data mining, Bayesian classifier becomes an effective tool to deal with the above uncertain classification problem with large amount of data in recent years.

At present, there are many ways to construct Bayesian classifiers. The Naive Bayesian classifier (NB) represents the simplest type of Bayesian network structure which estimates the probabilities using a fully independent model that incorporates all the problem variables [8]. A deficiency in NB is that there is an independent assumption of the variables, which is invalid in most cases and leads to predictive inaccuracies. As a breakthrough that allows dependencies among variables in the learned network, the Tree Augmented Naive Bayesian classifier (TAN) is an extension of the NB that uses a poly-tree structure defined over the set of feature variables to model the dependencies among feature variables not captured by the class variable [9]. However, the TAN allows each feature variable to be dependent with no more than one feature node and leads to the loss of some dependent information among all variables. Take such issues into account, Kazuo et al. developed an Advanced Pattern Recognition and Identification (APRI) system that automatically selects relevant variables and dependencies to build conditional probability models for classification problems by using the entropy-based concept of mutual information [10]. Recently, some novel Bayesian classifiers have been constructed by using some new techniques and algorithms such as data mining and artificial intelligence [11].

In this paper, a new Bayesian classifier model is proposed that employs genetic algorithms to determine the dependencies between the feature variables represented in the networks. When the model is applied to a customer predic-

*This research was supported by the Doctoral Foundation of China University of Petroleum. Corresponding author: hmshao@hdpu.edu.cn

978-0-7695-3304-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICNC.2008.724

tion problem, it matches potential churning customers well from the large amount of validation customers and shows higher classifying precision than the other three Bayesian classifier models. The rest of this paper is organized as follows. Some background of Bayesian classifiers is presented in Section 2 and our GA-based Bayesian classifier (GA-B) is constructed in Section 3. In Section 4, we apply the GA-based model to predicting churning customers and compare the performance of the 4 models: the NB, the TAN, the APRT and the GA-B. A summary of our work consists of Section 5.

2. Bayesian classifiers

A Bayesian network encodes the joint probability distribution of a set of N variables $(X_1, X_2, \cdots, X_N)$, as a directed acyclic graph (DAG) and a set of conditional probability tables (CPTs). Each node corresponds to a variable X_i and the CPT contains the probability of each state of the variable given every possible combination of states of its parents π_i.

Bayesian classifier is one of the simplest model of Bayesian networks that theoretically provides optimal classification performance for a given classification problem. The idea of such a model is to minimize the error rate by classifying new data according to the posterior probabilities of the various classes. Take a two-class problem for an example, the example X is assigned to class π_1 instead of class π_2, whenever $P(\pi_1|X) > P(\pi_2|X)$. If we had the joint distribution $P(\pi, X)$, we could readily compute the desired conditional probabilities. Of course, computing the joint distribution directly will frequently be infeasible because its size is exponential in the number of variables it references. In those cases, we must approximate the joint distribution by exploiting assumptions about its structure, which leads to the birth of all kinds of Bayesian networks.

3. Construction of Bayesian classifiers with GA

We consider a two-class problem in this section. Given a set of training data, where each data represents a customer's information, we obtain a set of attributions variables $(X_1, X_2, \cdots, X_N)$. Since this is a two-class problem, there is only one objective node determining which class will be assigned to for each feature X_i. In this section, we try to construct a Bayesian classifier that employs genetic algorithms to determine the dependencies between such feature variables, in hopes of possessing high true match rate and low false match rate.

A genetic algorithm is an adaptive heuristic search algorithm that supposedly simulates biological evolution based on the evolutionary ideas of natural selection and genetics [12]. In GA, each generation consists of individuals that are analogous to the chromosomes in our DNA. Each individual represents a possible solution in a search space and all the individuals in the population are then made to go through a process of evolution. The GA maintains a population of n chromosomes with associated fitness values and then parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan. Consequently highly fit solutions are given more opportunities to reproduce.

In our examined problem, a Bayesian network (BN) contains two kinds of nodes: one classification node and several attribution nodes. The following figure gives an example of a simple Bayesian network with one class node P and six attribution nodes.

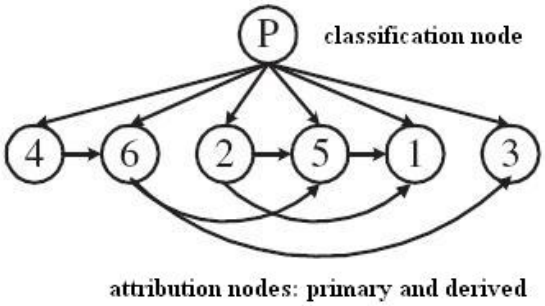

Figure 1. An example of Bayesian networks

There are two approaches to build a BN with GA:

- *attribution nodes are fixed and the BN structure is find by GA*
- *both of the BN structure and attribution nodes are selected by GA*

In this paper, we just give the first approach.

3.1. Chromosome Coding

Chromosome is the basic element of a generation. A fixed-length 3 dimensional array will be used as the chromosome, where the first array is used to represent the permutation of attribution nodes and the other two arrays to represent the possible father nodes for each attribution node.

1	2	3	4	5	6	7	8	9	Index
6	2	8	1	5	3	9	4	7	permutation of attributions nodes
*	1	1	3	*	5	*	6	2	first father node
*	*	2	1	*	*	2	4	*	second father node

Figure 2. An example of chromosomes

There are some restrictions on the network structure:

- *the classification node is the father node to all attribution nodes*
- *in-degree for each attribution node is not larger than 3*

The feasible values for each array is given below:

1. The attribution ID starts from 1. For a n attribution problem, feasible value for each position in the array is randomly selected within $\{1, 2, \ldots, n\}$ with uniform distribution.
2. The feasible number for the first link array satisfies the condition: for a given position with index t randomly select within $\{0, 1, 2, \ldots, t-1\}$ with uniform distribution, where 0 means no father node.
3. The feasible number for the second link array satisfies the conditions above so that it should not be identical with the same position in the first link array.

In this way, we can avoid circles in the DAG representation of Bayesian networks. Moreover, it can guarantee no more than two father nodes for each attribution node, except the class node.

3.2. Crossover Operator

Suppose we have the following two parent chromosomes as shown in Figure 3. There are two different types of array

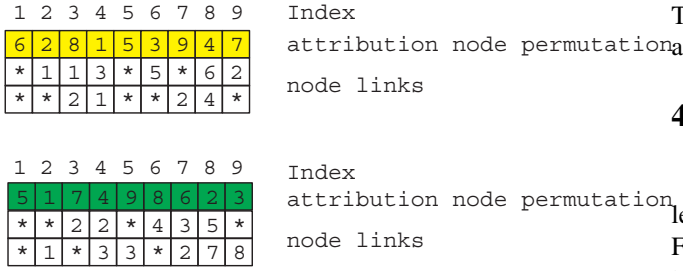

Figure 3. An example of two parent chromosomes

in a chromosome: node permutation and node links. In the process of Crossover, we use PMX crossover method for node permutation array and two-cut-point crossover method for node link, respectively.

3.3. Mutation Operator

In the next step, mutation operator will work on node permutation array and node link array separately. The following three types of mutation are used in this experiment.

- *Swap Mutation: swap two nodes ID in node permutation array*
- *Single Link Alter Mutation: alter node links for one node ID selected randomly*
- *Uniform Link Alter: alter node links for nodes selected randomly with uniform distribution*

Swap mutation method just works on node permutation array to yield a minor adjustment of the BN structure and Single Link Alter mutation method works on node link array, which just selects one node randomly and then recreates its link relationship to yield a minor adjustment of the BN structure. As to the Uniform Link Alter mutation method, it works on node link array, which just selects several nodes randomly with uniform distribution and then recreates their link relationship to yield a dramatic changes in BN structure.

3.4. Selection Operator

Selection is a major operator which mimic Darwinian natural selection to choose better rules from parent rules and child rules into next generation. At present, there are several ways to implement the process. In our problem, we adopt the *Roulette Wheel* selection and the F-score as the fitness function f. For each rule with fitness value f_k, its selection probability p_k is calculated as follows:

$$p_k = \frac{f_k}{\sum_{j=1}^{pop-size} f_j}$$

Then we can make a wheel according to these probabilities and select a rule for the next generation.

4 Performance comparisons

Once the structure is consulted, our classification problem is converted into a mathematic computation process. For sake of description, we denote the class node as the letter π and the attribution nodes as $X_1, X_1, \cdots, X_N$. Since this is a two-class problem, a customer is identified as loyal or churning and the value of class node π is denoted by π_1 and π_2. In our experiment, the threshold is set to be 0.5. That is, the customer will be assigned to be a churning customer if there holds the inequality

$$P(\pi = churn | X_1, X_1, \cdots, X_N) > 0.5$$

where

$$\begin{aligned} & P(\pi = \pi_i | X_1, X_1, \cdots, X_N) \\ &= \frac{P(\pi_i, X_1, X_1, \cdots, X_N)}{P(X_1, X_1, \cdots, X_N)} \\ &= \frac{P(\pi_i) P(X_1, X_1, \cdots, X_N | \pi_i)}{P(X_1, X_1, \cdots, X_N)} \end{aligned} \quad (1)$$

Based on different classifiers, the final expression of $P(\pi = \pi_i | X_1, X_1, \cdots, X_N)$ may differ from each other. In the next step, we compare the classification results and predictive accuracy of the following 4 models by using the same training data and validation data.

- *the native Bayesian classifier (NB)*
- *the tree-augmented native Bayesian classifer (TAN)*
- *the APRI classifier based on mutual information (APRI)*
- *the GA-based Bayesian classifier (GA-B)*

In the APRI classifier based on mutual information, we set the variable-selection threshold $T_{\pi x} = 1$ and the field-to-field threshold $T_{\pi x} = 0.1$. In the GA-based Bayesian classifier, the population is set to be 200 and the generation to be 1000.

4.1. Description of data sets

The data in this experiment is supplied by a Japanese credit debit and credit company, which is randomly selected from some current customers. The research aims to use the GA-based Bayes classifier system constructed in the last section to distinguish the potential churning customers based on some historical information. To evaluate its performance, we divide all the data into two parts: the training data and the validation data. After data preprocessing, 32 nodes are extracted as attribution nodes.

Table 1 shows the info of the training data and valadation data.

Table 1. Training data and validation data

\	Total	Churn set	Loyal set
Training data	748,158	34,768	713,390
Validation data	819,649	35,681	783.968

The data used to learn each model consists of more than four million records described by 32 variables. These variables represent a mixture of transaction-detail information and customer-summary information. In the validation data, churning customers make up 4.53% of all the population.

4.2. Result analysis

As usual, The decision made by the classifier can be represented in a confusion matrix (see Table 2). The confusion matrix has four categories: True positives (TP) are examples correctly labelled as positives. False positives (FP) refer to negative examples incorrectly labelled as positive. True negatives (TN) correspond to negatives correctly labelled as negative. Finally, false negatives (FN) refer to positive examples incorrectly labelled as negative [13].

Table 2. Confusion Matrix

\	Actual positive	Actual negative
Predicted positive	TP	FP
Predicted negative	FN	TN

In our application, the true positives are the churning customers correctly classified as the churning ones; the false positives are the loyal customers incorrectly classified as the churning ones. Table 3 presents the true match results based on validation set.

Table 3. True match of 4 models

$Models$	Churning customers	TP	TP rate
NB	59,518	17,890	50.14%
TAN	52,019	16,192	45.38%
APRI	50,261	17,648	49.46%
GA-B	46,195	18,992	53.23%

In this table, Columns 1-3 represent the model types, the predicted churning customers and the true identified customers. The true positive rate is listed in Column 4. From Table 3, we see that the GA-based Bayesian classifier has the most true match customers $(18,992)$ and the highest true match rate (53.23%) among the validation customers. The NB takes second place with the true match rate being up to 50.14%. As to the APRI on mutual information and the TAN, their true match rates are relatively somewhat lower, being 49.46% and 45.38%, respectively.

The comparison of the false match results by 4 models is shown in Table 4. From Table 4, we can see that the false

Table 4. False match of 4 models

$Models$	Churning customers	FP	FP rate
NB	59,518	41,628	5.13%
TAN	52,019	35,827	4.57%
APRI	50,261	32,613	4.16%
GA-B	46,195	27,203	3.47%

match rate of the GA-based Bayesian classifier is 3.47%, ranking the lowest of the 4 classifiers. The NB has the highest false match rate (5.13%). As to the TAN and the APRI, both of them behave better than the NB but worse than the the GA-B, with the false matching rate being 4.57% and 4.16%, respectively.

These comparisons refer to the correct classification of churning customers and incorrect classification of loyal ones. As we know, it is not enough to maximize the number of correctly classified churning customers; a good model

must simultaneously minimize the number of incorrectly classified loyal ones. This suggests examining the volume ratio given in the fourth column of Table 5, the ratio of incorrectly classified loyal customers to correctly classified churning customers (classification cost for short). An ideal model should minimize this ratio.

Table 5. Classification cost of 4 models

$Models$	TP	FP	Classification cost
NB	17,890	41,628	2.33
TAN	16,192	35,827	2.21
APRI	17,648	32,613	1.85
GA-B	18,992	27,203	1.43

The fourth volume of Table 5 tells us that the GA-B classifier has the lowest classification cost and the NB has the highest. Moreover, we see from Tables 4 and 5 that the GA-based Bayesian classifier has the most true match customers and the least false match customers. All of the above comparison results indicate that the GA-B has the highest classification accuracy than the others.

Despite of the above comparisons on false and true match, there isn't a uniform evaluation criterion to decide which classifier is good or bad. In most cases, it lies on the client's intention and preference. If he takes more care about true identification rate, the requirement of false match rate can be loosened to some extent. Contrarily, he will prefer an lower false match rate if the false match may lead to catastrophic cost.

5. Conclusion

In this paper, an effective GA-based Bayesian classifier model is constructed. To evaluate its performance, we have applied the model to a prediction problem of customer churn and compared the experiment results with the other three classifiers: the NB classifier, the TAN classifier and the APRI classifier based on mutual information. Experiment results indicate that the GA-based method identifies potential churning customers better from the large amount of validation customers and shows higher classifying precision compared with the others. Another superiority of the GA-based Bayesian classifier is that we can select different fitness functions or adapt the parameters of the fitness function to meet the needs of the clients with different preference.

References

[1] D. Van den Poel, B. Lariviere. Customer attrition analysis for financial services using proportional hazard models. European Journal of Operational Research, 157(1): 196-217, 2004.

[2] P.M. Dawkins, F.F. Reichheld. Customer retention as a competitive weapon. Directors-Board, Summer, pp.42-47, 1990.

[3] R. Ahmad, F. Buttle. Customer retention: a potentially potent marketing management strategy. Journal of Strategic Marketing, 9(1): 29-45, 2001.

[4] J. Ganesh, M.J. Arnold, K.E. Reynolds. Understanding the customer base of service providers: An examination of the differences between switchers and stayers. Journal of Marketing, 64(3): 65-87, 2000.

[5] C.B. Bhattacharya. When customers are members: Customer retention in paid membership contexts. Journal of the Academy of Marketing Science, 26(1): 31-44, 1998.

[6] M.R. Colgate, P.J. Danaher. Implementing a customer relationship strategy: The asymmetric impact of poor versus excellent execution. Journal of the Academy of Marketing Science, 28(3): 375-387, 2000.

[7] B. Lariviere, D. Van den Poel. Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Systems with Applications, 29: 472-484, 2005.

[8] P. Langley, S. Sage. Induction of Selective Bayesian Classifiers. Proc. 10th Conf. Uncertainty in Artificial Intelligence, Morgan Kaufmann, 399-406, 1994.

[9] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. Machine Learning, 1997.

[10] J.E. Kazuo, W.N. Steven. Constructing Bayesianian networks to predict uncollectible telecommunications accounts. IEEE Expert: Intelligent Systems and Their Applications, 11(5): 45-51, 1996.

[11] M.L. Wong, K.S. Leung. An efficient data mining method for learning Bayesianian networks using an evolutionary algorithm-based hybrid approach. IEEE Trans. on Evolutionary Computation, 8: 378-404, 2004.

[12] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. MA: Addison-Wesley, 1989.

[13] J. Davis, M. Goadrich. The relationship between precision-recall and ROC curves. In the Proc. of the 23rd Int. Conf. on Machine Learning (ICML), Pittsburgh, PA, 2006.