A Hybrid KNN-LR Classifier and its Application in Customer Churn Prediction

Yangming Zhang, Jiayin Qi, Huaying Shu, Jiantong Cao

Abstract— **This paper presents a hybrid approach for building a binary classifier. The approach is the combination of the k-nearest neighbor algorithm, handling separately m 1-dimensional data sets divided from a data set in m-dimension, and the logistic regression method. This hybrid KNN-LR classifier improves the performance of the logistic regression in classification accuracy in some situations where the predictor and target variables exhibit complex nonlinear relationships. The results of the experiment on four benchmark data sets show the proposed approach compares favorably with the well-known classification algorithms such as C4.5 and RBF. Furthermore, its effectiveness is illustrated by its application in customer churn prediction based on real-world customer data sets.**

I. INTRODUCTION

Logistic regression (LR) is a parametric approach for building binary classification model which is a core data mining task. It attempts to distinguish two classes from each other using a weighted sum of some predictor variables [13]. Theoretically speaking, its benefits include a firm statistical foundation and a probabilistic model useful for "explaining" the data. Meanwhile, in practice, Logistic regression has been shown to be extremely competitive with other classifiers in comparative studies [1].

LR is considered as a generalized linear regression. It allows one to predict a discrete outcome from a set of variables that may be continuous, categorical, or dichotomous. However linear regression is not appropriate for data with dichotomous outcomes in {0, 1}. Vulnerabilities in ordinary linear regression also impact logistic regression. Firstly, probability of target variable Y is a logistic function of the predictor variables X, the relation being assumed as monotonous. This hypothesis, however, is not always proper in practice, which may deteriorate classification accuracy. Secondly, LR needs categorical variables pretreatment. Each categorical variable in the model is converted to a vector of dichotomous variable in {0, 1}. That is, if a categorical variable x takes m values and k ($k < m$) distinct values $\{c_1, c_2, ..., c_k\}$ occurring in a training data set, it is replaced by a $k-1$ dummy variables d_1, d_2, …, d_{k-1}, such that $d_i = 1$ if $x = c_i$ and $d_i = 0$ otherwise, for $i = 1, ..., k-1$. If $x = c_k$, these dummy variables are zero. When an x's value in a data set is out of the set $\{c_1, c_2, ..., c_k\}$, it will be treated as c_k compulsively. In these cases, stabilization and accuracy of logistic regression model are jeopardized.

To overcome these weaknesses, the hybrid models combining logistic regression and neural networks have presented [14],[15]. They are, however, being criticized for long training process and limitation on understanding of and insight into the data. Therefore, a novel hybrid classifier, combining k-nearest neighbor (KNN) and logistic regression , is proposed in this paper. This hybrid KNN-LR classifier splits the classifier building process into two phases. In phase I, KNN is applied to each predictor variable separately, figuring out the proportion of positive instances to k number of instances closest to the query point in each one-dimension predictor. In phase II, LR with new predictor variables is trained by the data set transformed after phase I.

The paper is organized as follows. In Section II, LR and KNN are introduced briefly, followed by a detailed description of the hybrid KNN-LR classifier in Section III. Then we demonstrate the classification accuracy of the KNN-LR classifier in comparison with several typical binary classifiers on benchmark data sets in Section IV. In Section V, a customer churn prediction model built by KNN-LR is introduced. Finally, we discuss some prospects for future research.

II. RESEARCH METHODOLOGY AND LITERATURE REVIEW

A. Logistic Regression

Let X, y be a data set with dichotomous outcomes. For each instance x_i in X, the outcome is either $y_i = 1$ or $y_i = 0$. Instances with outcomes $y_i = 1$ are said to belong to the positive class, while instances with $y_i = 0$ belong to the negative class. We wish to create a regression model that identifies classification of an instance x_i as a positive or

Manuscript received March 31, 2006. This work was supported in part by the National Natural Science Foundation of China (Project No.: 70371056) and Information Management and Information Economy Key Lab of Ministry of Education of the People's Republic of China (project No.: 0607-41).

Yangming Zhang is with the School of Economics and Management, University of Posts and Telecommunications, Beijing, 100876 CHINA (e-mail: minglezhang@gmail.com).

Jiayin Qi is with the School of Economics and Management, University of Posts and Telecommunications, Beijing, 100876 CHINA (e-mail: ssqjy@263.net).

Huaying Shu is with the School of Economics and Management, University of Posts and Telecommunications, Beijing, 100876 CHINA (e-mail: Shuhy@bupt.edu.cn).

Jiantong Cao is with the School of Economics and Management, University of Posts and Telecommunications, Beijing, 100876 CHINA (e-mail: tony000@263.net).

negative class. The logistic function is introduced to establish a closed-form dependency between the probability of a class membership and the set of attributes. Its expression gives the following

$$P(y=1 \mid X,\beta)=\frac{e^{\beta X}}{1+e^{\beta X}} \quad (1)$$

Given a data set and a dependency from (1), the optimal set of coefficient β is determined by maximizing

$$\prod_{i=1}^{n} P(y_i \mid x_i,\beta) \quad (2)$$

Or in a formal mathematical notation

$$\begin{aligned}\beta^* &= \arg\max_{\beta}\{\prod_{i=1}^{n} P(y_i \mid x_i,\beta)\} \\ &= \arg\max_{\beta}\{\sum_{i=1}^{n} y_i \log(\frac{1}{1+e^{-\beta X}})+(1-y_i)\log(\frac{1}{1+e^{-\beta X}})\}\end{aligned} \quad (3)$$

Determining optimal coefficient β^* is called training. For an unseen instance x, and an optimal set of coefficient β^*, it can be classified after calculating the (1). If $P(y=1 \mid x,\beta^*)\geq 0.5$, the instance x should be labeled as positive, otherwise, it be labeled as negative.

B. KNN

K-nearest neighbor is a supervised learning algorithm where the result of a new instance is classified based on the majority of K-nearest neighbor category. This classifier does not use any model to fit and is only based on memory. Suppose that all instances with m attributes are labeled to either of two classes, positive or negative class. Given a query instance x_q, k number of training samples closest to it, denoted by $N_k(x_q)$, are found. $N_k^{+}(x_q)$ and $N_k^{-}(x_q)$ indicate the sets of positive and negative instances in $N_k(x_q)$ separately. If $|N_k^{+}(x_q)|>|N_k^{-}(x_q)|$, x_q is labeled as positive class, otherwise label it as negative one. In the meanwhile, the KNN is easily adapted to approximating continuous-valued target functions. To accomplish this, we have the algorithm estimate the probability of a query instance labeled as positive class correctly using the following equation.

$$p(x_q)=\frac{|N_k^{+}(x_q)|}{|N_k(x_q)|} \quad (4)$$

III. The Hybrid KNN-LR Classifier

A binary classifier D is a mapping, $D: S^m \rightarrow [0,1]$, where S^m is the m-dimensional space containing real number, binary as well as category. For an vector $x \in S^m$, $D(x)$ can be considered as a probability function, whose value is the probability that x is labeled as positive class.

In this paper, we will find a D using a combination of KNN and LR. Suppose that K and L are the mappings made up by KNN and LR respectively, we take a new mapping, the composition of K and L, as D. That is

$$D = L \circ K$$

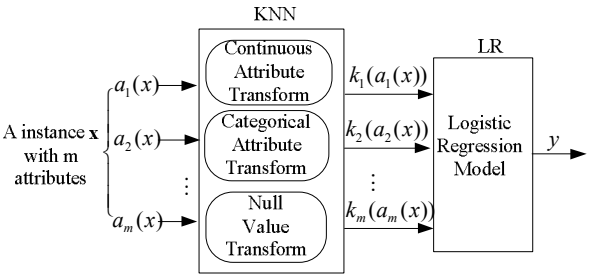

Fig. 1 Architecture of KNN-LR Classifier

Given a set of training data $X=\{x_1,x_2,\ldots,x_N\}$ in m-dimension and a set of labels $Y=\{y_1,y_2,\ldots,y_N\}\subseteq\{0,1\}$, our objective is to construct a hybrid classifier D, composed of a KNN classifier $K: S^m \rightarrow \Re^m$ and a LR classifier $L: \Re^m \rightarrow [0,1]$. This goal can be achieved through two phases shown in Fig. 1. In the first phase, K is trained by the data set X, Y, and then X is transformed into a new data set $K(X)$, a set of m-dimensional real vectors. In the new data set, any attribute value is real and related to the odds ratio linearly while it is real, dichotomous or categorical in the original data set. Moreover, a vector in $K(X)$ with a larger vector norm has a greater tendency towards positive class ($y=1$). In the second phase, a LR classifier L is trained by $K(X)$. A pseudo-code description of the KNN-LR training algorithm is sketched in Table I.

TABLE I KNN-LR Training Algorithm

1. Split training data set
$X \rightarrow \{a_1(X), a_2(X),\ldots,a_m(X)\}$
2. For each $a_i(X)$ do
$$k_i(a_i(x))=\ln\frac{|N_k^{+}(a_i(x))|}{|N_k^{-}(a_i(x))|}$$
3. Obtain new data set
$K(X)=(k_1(a_1(X)),k_2(a_2(X)),\ldots,k_m(a_m(X)))$
4. $P(y=1 \mid X,\beta)=\frac{e^{\beta K(X)}}{1-e^{\beta K(X)}}$ (5)
β^* is the estimation of β using (3).
5. output KNN-LR Model $P(y=1 \mid x,\beta^*)$

Let an arbitrary instance x be described by the attribute

vector $(a_1(x), a_2(x), \ldots, a_m(x))$, where $a_r(x)$ denotes the value of the rth attribute of instance x. Hereby, a set of training data X with m attributes can be divided into m data sets $\{a_1(X), a_2(X), \ldots, a_m(X)\}$ by attribute. KNN is employed to any split data set, in which instances correspond to points in the 1-dimensional space. Each KNN classifier related to a split data set can perform classification independently. Assuming the value the r[th] attribute of instances x is $a_r(x)$, we can calculate the probability of x labeled as positive class correctly from training data set $a_r(X)$ using (4). A real-valued target function $k_r(a_r(x))$, log odds rate of $p(a_r(x_r))$, has the form

$$k_r(a_r(x)) = \ln \frac{p(a_r(x_r))}{1 - p(a_r(x_r))} = \ln \frac{|N_k^+(a_r(x))|}{|N_k^-(a_r(x))|} \quad (5)$$

The definition of distance is the key to seek the nearest neighbors of an instance. The distance between the same attribute of two instances x_i and x_j is defined to be $d(a_r(x_i), a_r(x_j))$, where

$$d(a_r(x_i), a_r(x_j)) = |a_r(x_i) - a_r(x_j)|$$

for continuous-valued attributes,

$$d(a_r(x_i), a_r(x_j)) = \begin{cases} 0 & a_r(x_i) = a_r(x_j) \\ 1 & a_r(x_i) \neq a_r(x_j) \end{cases}$$

for categorical-valued attributes. Missing attribute values are assumed to be maximally different from the value present. If they are both missing, then $d(a_r(x_i), a_r(x_j))$ yields 0.

After unknown parameter β in (5) is estimated by the transformed training data set, the classification of any query instance can be identified by computing the following hybrid KNN-LR classifier, which is given by

$$P(y = 1 | X, \beta^*) = \frac{e^{\beta^* K(X)}}{1 - e^{\beta^* K(X)}} \quad (6)$$

IV. Experimental Results

We have used four data sets for training and validating our methodology. These are all real data sets, named, ADULT [11], CREDIT [11], TELECOM [12], and Wisconsin breast cancer [11]. All of them have two classes.

A. Data Sets

(1) ADULT: This census income database was donated by Ron Kohavi to the UCI repository. The task is to predict whether income exceeds \$50K/yr based on census data. We have selected a subset of 14 variables with the main goal to reduce the complexity of the problem but not with specific attention towards the predictive power of a variable.

(2) CREDIT: This data set was donated by J.R. Quinlan and can be obtained from the UCI repository or from the StatLog project under the name Australian Credit Approval. The goal is to predict credit approval. No detailed information is available about the meaning of the input variables, and all attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.

(3) TELECOM: This is a commercial application described in Weiss and Indurkhya (1995). The data describe a telecommunication problem and can be found at http://www.cs.su.oz.au/~nitin. They are also available from the RT homepage. In order to obtain a classification task we discretized the continuous output into class 0 for $y = 0$ and class 1 for $y \geq 0$. All predictor variables are continuous.

(4) Wisconsin breast cancer (BCW): This is one of the breast cancer databases at UCI, collected by W. H. Wolberg, the University of Wisconsin. The problem is to predict whether a tissue sample taken from a patient's breast is malignant or benign.

These four data sets are summarized in Table II.

TABLE II CHARACTERISTICS OF THE DATA SETS

Name of Data Set	No. of original attributes			Size of the data set and class distribution
	N.	C.	Tot.	
ADULT	6	8	14	48842(37155+11687)
CREDIT	7	8	15	690(307+383)
TELECOM	48	0	48	15000(9342+5658)
BCW	9	0	9	699 (458+ 241)

(N.: Continuous-valued attribute; C.: Categorical-valued attribute)

B. Evaluation Metrics

We compare performance using two evaluation metrics. First, we use classification accuracy: the number of correct predictions on the test data divided by the number of test instances. This has been the standard comparison metric used in studies of classification algorithms.

Classification accuracy obviously is not an appropriate evaluation criterion for all classification [5]. For this work we also want to evaluate and compare different methods with respect to their estimates of class probabilities. One alternative to classification accuracy is to use ROC (Receiver Operating Characteristic) analysis [6], which compares visually the classifiers' performances over the entire range of probabilities. For a given binary classifier that produces a score indicating likelihood of class membership, its ROC curve depicts all possible tradeoffs between true-positive rate (TP) and false-positive rate (FP). Specifically, any classification threshold on the score will classify correctly an expected percentage of truly positive cases as being positive (TP) and will classify incorrectly an expected percentage of negative examples as being positive (FP). The ROC curve plots the observed TP versus FP for all possible classification thresholds. Objective comparisons can be made with ROC analysis. However, we want to evaluate the class probability estimates generally rather than under specific conditions or under ranges of conditions. In particular, we concentrate on

how well the probability estimates can rank cases by their likelihoods of class membership. Therefore, the area under the ROC curve (AUR) [7], a metric for comparing classifiers across a wide range of conditions, is introduced. AUR measures the quality of an estimator's classification performance, averaged across all possible probability thresholds.

In this paper, we report the classification accuracy and AUR when comparing class probability estimation.

C. Results

In the comparative study, we compare it in classification accuracy with LR, C4.5 [8], [9] and radial basis function (RBF) network [10]. We have implemented all of them in Java, while codes of the latter three comparative algorithms come from WEKA machine learning project (found at http://www.cs.waikato.ac.nz/~ml/). C4.5 runs with the default settings in WEKA including pruning. To obtain probability estimates from these trees we used the frequency scores at the leaves. For RBF, we construct a network with only one hidden layer. Berry and Linoff suggest the hidden layer should never be more than twice as large as the input layer [16]. In this study, we set the number of hidden nodes according to a general rule, (number of inputs + outputs) * (2/3), from the FAQ for a commercial neural network software company. An improvement in KNN-LR can be gained by optimizing the value of k. We set k=15 in these experiments. All experiments in this paper are ten-fold cross-validations. The results are shown in Table III and Table IV.

TABLE III AUR COMPARISON

	KNN-LR	LR	C4.5	RBF
ADULT	0.917	0.904	0.889	0.884
CREDIT	0.923	0.902	0.879	0.876
TELECOM	0.977	0.931	0.983	0.951
BCW	0.990	0.992	0.937	0.985

TABLE IV CLASSIFICATION ACCURACY COMPARISON

(%)	KNN-LR	LR	C4.5	RBF
ADULT	86.34	85.17	86.15	84.03
CREDIT	86.38	84.93	85.51	81.45
TELECOM	92.55	86.71	97.67	88.30
BCW	96.14	96.28	93.99	95.28

As expected, KNN-LR outperforms LR on these benchmark data sets, except for that they have close performance on the BCW data set. The results also show its superiority over C4.5 and RBF, on behalf of decision trees and neural networks classification algorithms, in our experiments. C4.5 just beats KNN-LR on TELECOM, in which have 48 continuous-valued attributes, because decision trees and neural networks are generally dominant in situations where the predictor and target variables exhibit complex nonlinear relationships. However, KNN-LR, which preprocesses instances attributes using KNN before applying LR, achieves higher than the original LR 5.84% in classification accuracy and 0.046 in AUR, and even outperforms RBF on the TELECOM data set. The experimental results prove KNN-LR performs better than LR on classification precision in complex situations.

V. APPLICATION IN CUSTOMER CHURN PREDICTION

Customers become "churners" when they discontinue their subscription and move their business to a competitor. Customer retention is important to a company since it costs 5~10 times more to recruit a new customer than to retain an existing one, especially in the mobile telecommunications industry where competition is fierce. In order to support mobile carriers to reduce churn rate, we need to predict which customers are high risk of churn and optimize their marketing intervention resource to prevent as many customers as possible from churning. The effect of customer retention strategies depends upon the prediction model's accuracy. In our study, the KNN-LR classifier is introduced to predict the probability of customer churn.

A. Data Source

Information about mobile customers such as user demographics, contractual data, customer service logs, bill records and call details are collected from a mobile carrier in China. Specifically, the call details in August and September, 2003 and bill records from February to July, 2003 are available in this evaluation. Customers' status in a month is defined as churner or non-churner according to their relationship with the mobile carrier. That is, churner is the customer who cancels his/her with the mobile carrier on his/her own initiative. Otherwise, the customer is non-churner. All the customers investigated in this study were extracted from customer base randomly. The samples are divided into two groups, one is the training data set consisting of 40,000 non-churners and 2,000 churners, the rest 10,000 non-churners and 495 churners are for testing.

Through experts advising, we constructed 93 input variables associated with each customer that we conjectured might be linked to churn. These variables included

- customer demographics;
- contractual data, such as payment type, contract type, duration in service;
- monthly charges and usage;
- count of distinct phone numbers contacting within the closest 2 months;
- revenue from diverse services;
- number of active services;
- number of opening or terminating services within the closest 3 months.

B. Churn Prediction

Estimated by the churn model constructed, we obtain the probability of churn for each customer. The continuous probability measures must be thresholded to require a churn or non-churn prediction. For a given threshold on the probability of churn, we

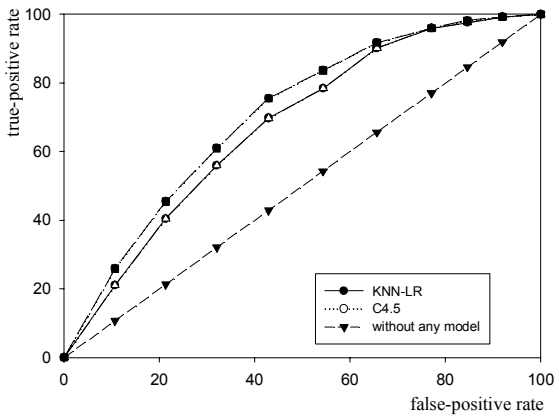

Fig. 2 KNN-LR vs. C4.5 in customer churn prediction

determine two quantities: 1) the fraction of all non-churners having churn probability above the threshold and 2) the fraction of all churners having churn probability above the threshold. These two quantities are corresponding to axes X and Y. The churners have the same churn probability to the non-churners without any churn prediction model. In this situation, the straight line shown in Fig. 2 indicates no discrimination of churners from non-churners. By contraries, the more bowed the curve is to the upper-left corner of the graph, the better the customer is at discriminating churners from non-churners.

VI. CONCLUSIONS

The KNN-LR method combining KNN and LR presented here provides an alternative approach for binary classification model. It improves the classification accuracy of original LR, especially in complex situations where the relationships between predictor and target variables are nonlinear. For the future work, we need to explore the validity of the proposed method in theory, and want to pretreat data with KNN in other ways which reduce negative effects on classification accuracy due to interactions among predictor variables.

REFERENCES

[1] T.S. Lim, W.Y. Loh, and Y.S. Shih. "A comparison of prediction accuracy, complexity, and training time for thirty–three old and new classification algorithms". Machine Learning, 40, pp.203–228, 2000.

[2] Tuhao Chen, Fred M. Hoppe, Satish Iyengar and David Brent, "A Hybrid Logistic Model for Case-Control Studies", Methodology and Computing in Applied Probability, Vol 5, Num 4 , pp.419–426, December 2003

[3] Aha, DW, D. Kibler & MK Albert, "Instance-Based Learning Algorithms", Machine Learning, 6, pp.37–66, 1991.

[4] S.M. Weiss and N. Indurkhya. "Rule–based machine learning methods for functional prediction", Journal of Artificial Intelligence Research, 3:383–403, 1995.

[5] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms", In Proceedings of the Fifteenth International Conference onMachine Learning, pp.445–453, Morgan Kaufmann, San Mateo, California, 1998.

[6] J. Swets, "Measuring the accuracy of diagnostic systems", Science, 240: pp.1285–1293, 1988.

[7] A.P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms", Pattern Recognition, 30: pp.1145–1159, 1997.

[8] J. R. Quilan, "C4.5: Programs for Machine learning", Morgan Kaufmann, San Mateo, CA, 1993.

[9] J. R. Quilan, "Improved use of continuous attributes in C4.5", Journal of Artifical Intelligence Research, 4: pp.77-90, 1996

[10] Bishop, C. M., "Neural networks for pattern recognition", Oxford, England: Oxford University Press, 1995.

[11] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases", 1998.

[12] S.M. Weiss and N. Indurkhya, "Rule–based machine learning methods for functional prediction", Journal of Artificial Intelligence Research, 3: pp.383–403, 1995.

[13] Pampel, F. C., "Logistic regression—A premier", Thousand Oaks, CA:Sage,2000.

[14] TS Lee, CC Chiu, CJ Lu, IF Chen, "Credit scoring using the hybrid neural discriminant technique", Expert Systems with Applications, Vol 23,pp.245–254,2002.

[15] Y Bentz, D Merunka, "Neural networks and the multinomial logit for brand choice modelling: a hybrid approach", Journal of Forecasting, Vol 19, pp.177–200, 2000.

[16] Berry, M.J.A., and Linoff, G., Data Mining Techniques, NY: John, pp. 323, 1997