An effective hybrid learning system for telecommunication churn prediction

Ying Huang *, Tahar Kechadi

School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland

1. Introduction

With recent evolution in the Information and Communication Technology (ICT) sector, numerous new and attractive services have been introduced, and they put huge pressure on traditional services. Customer churn has emerged as one of the major issues in Customer Relationship Management (CRM) in telecommunication services around the world, for both wireless providers and long-distance carriers. For instance, in the U.S., telecom providers of long-distance and international services have been bearing the churn rates from 45% to 70% percent for some years (Mattison, 2001). Under the fierce competitive environment, it becomes very important for the telecom operators to retain their existing customers as acquiring new customers is much more expensive. Consequently, predicting which customers are likely to stop their subscription and switch to competitors (churn) is critical. Predicting the potential churners and successfully retain them, especially the valuable ones, can substantially increase the profitability of a company.

In the telecommunications industry, operators usually capture the transactional data, which reflects the service usage, and some static data such as subscriber’s personal information and contract details. Data mining (DM) methods have emerged as a good alternative to study the customer behaviour. We can find various DM techniques, such as decision tree, logistic regression, support vector machine, artificial neural networks, inductive rule learning, etc. They have been applied to predict customer behaviour (Huang, Huang, & Kechadi, 2011; Hwang, Jung, & Suh, 2004; Larivire & Poel, 2005; Wei & Chiu, 2002; Xia & Dong, 2008). Most of the existing predictive modelling techniques, applied to customer churn, are based on supervised learning; very few of them have been based on unsupervised learning. In addition, most of the classifiers use single model (i.e., only one data mining technique). Many of the single model-based classifiers can predict potential churners to a large extent. However, either the accuracy is not good enough for some of the techniques or there is a room for improving the prediction accuracy for some others, and a hybrid model is a good alternative for better classification performance. Moreover, usually the entire training data instances are all used to build prediction models. However, it may be more effective to predict a new data instance based on partial training instances that are more similar to the test data than other training instances.

The advantages of the proposed model over the other commonly used modelling techniques in the domain of churn prediction concern the following aspects: Firstly, the prediction model...
of the proposed system is a hybrid-model, which fully integrates supervised and unsupervised learning by employing a clustering method for dividing the training data and a rule induction method for generalising classification rules for each cluster. Secondly, the process for predicting a test data instance relies on the most appropriate sub-classifier, which is produced from the most similar cluster to the test instances rather than the entire training set. We argue that using only a sub-classifier may improve the prediction accuracy. Thirdly, the commonly used k-means algorithm is used for clustering the original training data; however, to enhance the clustering result, we apply a weighting technique, which makes full use of the causality between attributes and the target. The experimental results show that modified k-means enhances not only the clustering results but also the prediction results.

The hybrid-model based prediction system has four main phases, which are as follows:

1. **Data Discretisation**: We apply a class-dependent discretisation method on all the continuous attributes. The continuous data is transformed to the form of intervals, which is conducive to the rule induction.
2. **Weighted Clustering**: We apply a weighted k-means clustering algorithm to divide the training data into a number of clusters. We use Path Analysis to calculate the weights of attributes.
3. **Rule Extraction**: We apply a rule learning method (i.e., FOIL) to each cluster, extracted in the previous phase, to induce a set of classification rules, which constitutes a sub-classifier. Thus, each cluster corresponds to a sub-classifier.
4. **Prediction**: We predict each test instance by choosing the most suitable sub-classifier, which corresponds to the closest cluster to the test data, according to a distance or similarity measure.

The remainder of the paper is organised as follows: Section 2 reviews some related work in the area. Section 3 describes the details of the proposed hybrid learning system. The experimental set-up and results are discussed in Section 4. We conclude and outline some future work in Section 5.

### 2. Related Work

Large number of machine learning and knowledge discovery techniques have been proposed and applied to the problem of customer retention in the domain of CRM. The techniques can be used in different phases of the data mining process, such as eliminating the noise and outliers, reducing the feature space by selecting most relevant attributes, predicting churn behaviour, etc.

In this section, we briefly introduce ways of building a classification model by reviewing several customer churn prediction models proposed in the literature. Hung, Yen, and Wang (2006) developed two classification methods by using three models, one is based on k-means clustering and decision tree (C5.0) and the other combines the back-propagation neural network (BPN) and decision tree. The models are evaluated using LIFT and hit ratio measures. Huang, Kechadi, and Buckley (2012) proposed a mining process that consists of feature extraction and classification. The authors implemented seven traditional classification modelling techniques (e.g., Decision Tree, Linear Regression, and Multilayer Perceptron Neural Networks, etc.) to build different predictive models, and for some models, they use different data processing methods. They have evaluated the performance of their approach using true positive and false positive measures.

Apart from the techniques described above, in fact, there is a large amount of literature reporting on the application of data mining techniques to study the behaviour of customers in telecommunications, (Au, Chan, & Yao, 2003; Coussement & Den Poel, 2008; Huang et al., 2011; Mozer, Wolniewicz, Grimes, Johnson, & Kau-shansky, 2000). Ngai, Xi, and Chau (2009) reviewed more than 80 papers about the application of data mining to Customer Relationship Management, and a lot of them concern the domain of customer churn prediction. Nonetheless, most of the proposed modelling techniques use a single model.

Recently, many researchers have started to study hybrid models to improve the classification effectiveness. Usually, hybrid models combine two or more techniques. For instance, the classification or clustering techniques can be sequentially combined (e.g., Khashei, Hamadani, & Bijari (2012) proposed a hybrid classification model by integrating artificial neural networks and multiple linear regression to yield more general and accurate classification result than single traditional method). Tsai and Lu (2009) suggested that the hybrid techniques can provide better performance than many single model-based approaches in numerous different domains. One can refer to Lee and Lee (2006) for some common types and structures of hybrid model-based classification methods. The authors have built a hybrid model called SePI (Segmentation by Performance Information). The framework of SePI is based on three models, main model, discrimination model, and support model. Decision tree (C5.0), which found to be one of the best single model for a given data, is chosen as the main model; the discrimination model uses the performance information of the main model on the training dataset; the support model uses the data for which the main model predicted incorrectly, and ANN is employed as the support model. The key idea of SePI is that if the test examples cannot be predicted correctly by the main model, then they will be predicted by the support model.

For the customer churn prediction problem, most of the related work focuses on using only one data mining method. Table 1 shows the related implementations that employ hybrid modelling techniques for customer churn prediction. Most of the hybrid model-based applications normally follow a common pattern, which consists of two stages: the first stage deals with pre-processing the input data (e.g., reduce the dataset, detect the outliers, etc.), the second stage is dedicated to the mining of the preprocessed data to extract useful patterns. In our study, we also build a hybrid learning system by integrating two stages. However, the difference is that the first stage is no longer considered as the step for data pre-processing. We re-design the system by firstly segmenting the training data into different groups; we build a set of sub-classifiers by applying rule induction method on each group of the data; the second stage predicts the categories of the test examples, each example (e.g., test\(_i\)) is predicted by a sub-classifier that was produced by the closest sub-cluster to test\(_i\).
be a dataset. Let $d$ be a dataset. Let $D$ be a dataset. Let $C_i$ be a dataset. Let $K$ be the number of clusters.

### Table 1

<table>
<thead>
<tr>
<th>Hybrid system</th>
<th>Data set</th>
<th>Techniques</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SePI</td>
<td>Customer churn data of a Korean company; 11,587 subscribers’ demographic and transaction data; 24 features.</td>
<td>DT (C5.0), NN, LR Lee and Lee (2006)</td>
<td>10-fold cross validation, hit ratio classification accuracy, ROC analysis</td>
</tr>
<tr>
<td>KNN - LR</td>
<td>Customer data of a Chinese mobile carrier; training set (40,000 non-churners and 2,000 churners), testing set (10,000 non-churners and 495 churners); 93 features.</td>
<td>KNN, LR Zhang et al. (2007)</td>
<td>5-fold cross validation, overall prediction accuracy, two types of error assessment</td>
</tr>
<tr>
<td>(NN - NN) and (SOM - NN)</td>
<td>Dataset provided by American telecom company, including 34,761 churners and 16,545 non-churners from July 2001 to January 2002</td>
<td>NNs, SOMs Tsai and Lu (2009)</td>
<td>Lift Curve</td>
</tr>
<tr>
<td>Unsupervised techniques - Decision Tree with Boosting</td>
<td>Three sets of data are obtained from the Teradata center at Duke University, USA. Set1: 100,000 records, churn rate 50%; Set2: 50,000 records, churn rate 1.8%; Set3: 100,000 records, churn rate 1.8%</td>
<td>DT (C5.0) with boosting, clustering techniques (i.e., k-means, k-m oid, SOM, FCM, and BIRCH) Bose and Chen (2009)</td>
<td>5-fold cross validation, overall prediction accuracy, two types of error assessment</td>
</tr>
<tr>
<td>DT - GA</td>
<td>1.2 million subscribers Call Details Records from the largest telecom operator in a developing country during 1st July and 1st September 2009.</td>
<td>DT (C4.5), GA Yeshwanth et al. (2011)</td>
<td>Accuracy, False Positive</td>
</tr>
</tbody>
</table>

### 3.1. Discretization

The discretisation process is widely used in data mining and statistical analysis problems. It partitions continuous attributes into discretised intervals, and it is usually carried out as the first step for implementing probability functions or inductive learning. Therefore, the effectiveness of discretization contributes to the performance of the entire analysis. Many discrete methods have been developed and have shown to be effective (Ching, Wong, & Chan, 1995; Fayyad & Irani, 1993; Kurgan & Cios, 2004). The data used in this study contains a set of symbolic and continuous attributes. The continuous attributes are unsuitable for inductive rule learning. Therefore, we adopt the Class-Dependent discretisation technique for continuous attributes. This approach aims to maximize the dependency relationship (i.e., mutual dependence) between the class variables and continuous attributes, and automatically decide the number of intervals for an inductive learning application. More information about the process of discretisation can be found in Ching et al. (1995).

### 3.2. Clustering Algorithm

Clustering technique is widely used in many data mining applications. It groups data objects based on information that describes the objects and their relationships (Tan, Steinbach, & Kumar, 2005), called knowledge. In our approach we use clustering technique to segment the training data.

A number of different types of clustering methods, such as partitional clustering, hierarchical clustering, and fuzzy clustering, etc., have been studied in the last two decades or so. Among those clustering algorithms, k-means, which belongs to the family of partitional clustering, is very popular due to its simplicity. We adopted the core idea of this algorithm and proposed a weighted k-means clustering method to more effectively partition the data into groups with different characteristics, as the given application is very complex and its datasets are heterogeneous and containing a large number of attributes.

#### 3.2.1. K-means

Algorithm 1 illustrates the main procedure of the classic k-means clustering. Consider a dataset $D$, let $K$ be the number of clusters and $o$ be an object of dataset $D$, $C_i$ and $c_i$ denote the $i^{th}$ cluster and its centroid, respectively. The algorithm randomly selects $K$ initial centroids; then it iterates through the following steps; each
data object in \( D \) is assigned to its nearest cluster, this is followed by the recalculation of the cluster centroids. This algorithm stops when centroids are unchanged.

**Algorithm 1. K-means Algorithm**

1: Select \( K \) objects as the initial centroids;  
2: repeat  
3: Assign each object \((o)\) to its closest centroid \((C_i)\);  
4: Re-compute the centroids of all clusters;  
5: until The value of objective function (SSE) does not change.

The main goal of clustering is that the objects within a group are similar and objects belonging to different group are dissimilar. Therefore, one needs to define similarity or a distance measure between data objects. In this analysis we use the Euclidian distance. We use the objective function SSE (Sum of the Squared Error) to measure the quality of the clustering, and SSE is calculated as follows:

\[
SSE = \sum_{i=1}^{K} \sum_{j=1}^{n_i} ||(C_i - o_j)||^2
\]

where \( n_i \) is the number of data objects in cluster \( C_i \) and \( o_j \) is the \( j \)th data object in cluster \( C_i \). The smaller the value of SSE, the better is the quality of the clustering. The algorithm stops when SSE is minimised. More details about the K-means algorithm can be found in Han and Kamber (2000).

### 3.2.2. Weighted K-means

The datasets we are using are provided by a telecom company. These datasets have very large number of attributes, and each attribute has different effect on the target (i.e., class). Thus, it may be more effective to deal with each attribute differently. We assign a weight for each attribute in order to reduce high influence of an attribute on the target. We believe that this can help to extract the knowledge from the datasets more objectively and accurately. In order to study the influence of each attribute, we consider the causal relationship between attributes and the target.

Vasconcelos, Almeida, and Nobre (1998) found that the simple correlation coefficient-based analysis does not reflect the direct effect of independent variables upon the dependent variable. Therefore, the correlation coefficient analysis usually tends to draw a wrong conclusion. Furthermore, multiple linear regression investigates the correlation relationship between multivariate variables and dependent variables, but it does not figure out the joint effect caused by multiple variables. Thus, to quantify the significance of attributes, we apply Path Analysis (PA) to discover the causality between variables. PA, which is a multivariate statistical analysis, provides not only the direct effect of an independent variable on a dependent variable but also the indirect effect of an independent variable on a dependent variable via other independent variables. PA has been applied in various fields of social science, such as public health psychology, agriculture, business, etc., one can refer to the applications in Duncan (1966), Scheiner, Mitchell, and Callahan (2000) and Vasconcelos et al. (1998). More details on PA can be found in Alwin and Hauser (1975) and Denis and Legerski (2006).

In order to analyse the causal effect of \( x = \{x_1, x_2, \ldots, x_n\} \) (independent variables) on \( y \) (dependent variable), PA decomposes the correlation coefficient \( r_{xy} \) into the direct effect of \( x_i \) on \( y \) and the indirect effect of the \( x_i \) on \( y \) via other independent variables (say \( x_j \)). The direct effect of \( x_i \) on \( y \) is quantified by Path Coefficient \( p_{xy} \). To calculate \( p_{xy} \), we define the correlation coefficient matrices (2) and (3) that represent the correlation matrix among variables \( x = \{x_1, x_2, \ldots, x_n\} \) and the correlation matrix between \( x \) and \( y \), respectively:

\[
R = \begin{pmatrix}
1 & r_{12} & \cdots & r_{1n} \\
r_{21} & 1 & \cdots & r_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
r_{n1} & r_{n2} & \cdots & 1
\end{pmatrix}
\]  

(2)

\[
r_{xy} = \begin{pmatrix}
R_{xy} \\
R_{yx}
\end{pmatrix}
\]

(3)

The direct path coefficient can be calculated as follows:

\[
p_{xy} = R_{xy}^{-1} R_{yx}
\]

(4)

In addition to domain experts, it is usually difficult to set appropriate paths, which reflect the way the independent variables affect the dependent variables. This is also why there is little literature about PA, as a multivariate analysis technique. In this paper, we started by considering that the indirect effect of \( x_i \) on \( y \) (called indirect path coefficient) by combining it with one variable \( x_j \) (\( j \neq i \)) only. In order to set more effectively the weights, we use another concept; the determinant factor \( (R_{ij}^2) \), which reflects the integrated determinant effects of \( x_i \) on \( y \). \( R_{ij}^2 \) is calculated as follows:

\[
R_{ij}^2 = 2p_{xy}R_{xyn} - p_{xy}^2
\]

(5)

We define the weight of each attribute as the ratio of the determinant factor \( (R_{ij}^2) \) over the sum of all determinant factors, (see the Eq. (6)):

\[
w_i = \frac{R_{ij}^2}{\sum_{j=1}^{n} R_{ij}^2}
\]

(6)

Due to the high-dimensionality of the datasets used, we adopt the Minkowski distance of order \( p \), and we define \( p \) to be the total number of attributes. To reflect the influence of attributes, we integrate the weights in the distance measure as follows:

\[
dist = \left( \sum_{i=1}^{n} w_i \cdot (x_i - x_i')^p \right)^{1/p}
\]

(7)

where \( x_i \) and \( x_i' \) represent the attribute values of two different objects.

### 3.3. Inductive Rule Learning

Inductive rule learning is another core of this study. A set of rules can be generated from each cluster, and the rules can be used to represent the characteristics of the clusters. The main advantage of the rule-based learning is that the results can easily be understood and interpreted. In this research, we apply FOIL (First Order Inductive Learning), which is a classic sequential covering technique based algorithm, to learn about each group of the training data.

Sequential covering algorithm learns one rule at a time and removes any positive examples it covers, and then it iterates through this process until all positive examples are covered by the learned rules. FOIL is an extension of the sequential covering algorithm to the first-order representation. FOIL has been widely used by rule-based classification algorithms and techniques, such as Qin and Lawry (2005), Qin and Lawry (2008), Rai, Thoke, and Verma (2012) and Yin and Han (2003). These classification models implement pruning strategies or optimisation process on the basis of
FOIL's architecture. For the application of customer churn prediction, we are interested more on the behaviour of potential churners. Moreover, as FOIL is capable of sequentially generating a set of positive rules (i.e., churn rules in this application), we use it for rule-based learning. Algorithm 2 briefly outlines the FOIL algorithm, the input variable Example denotes the training set with labelled data examples.

**Algorithm 2. FOIL (Examples)**

1. Pos ← Positive Examples;  
2. Neg ← Negative Examples;  
3. Learned_rules ← ϕ;  
4. while Pos is not empty do  
5. Rule ← Learn-A-Rule (Examples, Neg);  
6. Learned_rules ← Learned_rules ∪ Rule;  
7. Pos ← Pos - {positive examples covered by Rule};  
8. Examples ← Examples - {any examples covered by Rule};  
9. end while  
10. return Learned_rules

**Algorithm 3. Learn-A-Rule (Examples, Neg)**

1. Rule ← the most general positive rule;  
2. repeat  
3. Candidate_condd ← generate candidate conditions for Rule;  
4. Best_condd ← max (Gain);  
5. add Best_condd to the antecedent of Rule;  
6. covered_Negd ← negative examples that are covered by Rule;  
7. until There is no negative examples can be covered.  
8. return Rule

More information of this algorithm, such as the calculation of information gain, can be found in Quinlan and Cameron-Jones (1993).

3.4. Classification

To predict whether a customer is about to leave or not, we use clustering together with the inductive rule learning approach described in Section 3.3. We argue that customers having similar characteristics also behave similarly. In the case of churn problem, customers (instances) who have similar patterns, for instance, having very close calls behaviour, spending almost the same, etc., are more likely to belong to the same class (churn or non-churn). Thence, we aim to predict each test instance (tci) by referring to the instances with similar behaviour (we call them Close Instances). The goal is to have all Close Instances in the same cluster (Close Cluster) to tci. Closest is quantified by the minimum distance measure between the instance and clusters, and the Minkowski distance (Eq. (7)) is used in this case. The final classifier (Close Sub-classifier) is produced by applying the rule learning method (FOIL) on the Close Cluster. Therefore, the classification of a tci is carried out by the following principles:

- If the Close Cluster is pure and all instances in Close Cluster are positive (churn), then tci is predicted as churn;
- If the Close Cluster is pure and all instances in Close Cluster are negative (non-churn), then tci is predicted as non-churn;
- If the Close Cluster is impure (having both churn and non-churn), then the class label of tci is assigned by the Close Sub-classifier, which consists of a set of positive rules. tci is predicted as churn if it was covered by any of the rules, otherwise, it will be assigned to the majority class (non-churn in this application).

In addition to the cases mentioned above, if the number of Close Clusters is more than one, that is, more than one clusters have the same minimal distance with tci, then tci is predicted by the majority prediction of all Close Sub-classifier. For example, three clusters (c1, c2, c3) are all Close Clusters, if the sub-classifiers generated by c1 and c2 predict tci to be non-churn, while the result churn is obtained from c3, then tci is labelled to non-churn.

4. Experiments and Discussion

4.1. Data

Experiments were conducted on a telecom dataset. The whole dataset consists of 104,199 customer records: 6,056 churners and 98,143 non-churners. Each customer record is characterised by 121 attributes, including 11 symbolic attributes and 110 continuous attributes. The attributes mainly consist of the following information:

**Demographic profiles:**

describes the demographic information, including age, gender, social class bands, county code, etc.

**Account information:**

total information about customer accounts, e.g., start time, account number/type, fees, payment type, account balance, call information, etc.

**Call details:**

this group of information describes many aspects related to different types of calls (e.g., international or local calls), number of calls, call duration, costs, etc.

Apart from the attribute information mentioned above, one can refer to Huang, Kechadi, and Buckley (2008) and Huang, Sato, Huang, Kechadi, and Buckley (2010a) for more other information about the data (e.g., Information of grants, service orders, historical information of payments, etc.).

4.2. Evaluation

In order to evaluate the performance of a predictive model objectively, we use 5-fold cross validation model. The data is randomly partitioned into five nearly equal size sets; during each iteration one set is used as the validation data, while the remaining four sets of data are used for training. In other words, the validation process is repeated five times; each of the five datasets is used exactly once as the validation data. The five validation results are then averaged to produce the final result.

Usually, a predictor is considered to be effective if a large number of instances can be correctly predicted. Thus, in much literature, the overall predictive accuracy (i.e., \(\text{num\_correct}/\text{num\_total}\)) is employed to evaluate the classification performance, such as Tsai and Lu (2009), Yeshwanth, Raj, and Saravanan (2011) and Zhang, Qi, Shu, and Cao (2007). However, the overall accuracy-based evaluation criterion does not suite imbalanced data (the telecom data in this study is one of them). For instance, for imbalanced data, a high overall inaccuracy might be caused by the fact that many data examples were
wrongly assigned to the majority class. Therefore, in this work, for better assessing the power of a classification technique, we adopt the rates of True Churn (TC), which indicates the proportion of churn examples that were correctly predicted as churn, and False Churn (FC), which is the proportion of non-churn examples that were wrongly predicted as churn. The objective of this application is to obtain high TC with low FC. Table 2 shows the confusion matrix, where the four parameters are defined:

<table>
<thead>
<tr>
<th>Actual</th>
<th>CHURN</th>
<th>NONCHURN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHURN</td>
<td>$a_{11}$</td>
<td>$a_{12}$</td>
</tr>
<tr>
<td>NONCHURN</td>
<td>$a_{21}$</td>
<td>$a_{22}$</td>
</tr>
</tbody>
</table>

4.3. Experiment set-up

In order to comprehensively investigate the proposed learning system, three sets of experiments were conducted. The first experiment is for evaluating the performance of the hybrid model and the second and third are for comparing it to well-known existing models.

4.3.1. Set-up I

The performance of this hybrid classification system highly depends on the result of clustering. Therefore, in our experiments, we compare the clustering results derived from the original and the modified $k$-means algorithm, respectively. In addition, to provide more accurate weight for each attribute, we also compare the weighted clustering approaches, which are based on the path analysis and other correlation analysis methods (i.e., Chi-Square and information gain).

Chi-Square can be used to evaluate the dependence between an input variable and the class, and it has been widely applied in many phases of data mining process. In this application, we calculate the Chi-Square statistic ($\chi^2$) for all attributes, $\{\chi_1^2, \chi_2^2, \ldots, \chi_m^2\}$, and the weight of an attribute (say the $i$th attribute), is defined by $\sqrt[2n]{\chi_i^2}$, where $n$ is the total number of attributes.

In addition to Chi-Square, information gain is also used to measure the significance of attributes. Similar to the weighting methods that are based on path analysis and Chi-Square, we calculate the information gain of all attributes, $\{g_1, g_2, \ldots, g_m\}$, and the final weight of each attribute is defined as the fraction of $g_i$ by the sum of the information gain of all attributes, which is $\sum_{i=1}^{n} g_i$.

4.3.2. Set-up II

In the second set of experiments, we compare the classification performance of the hybrid learning system with six other different types of supervised classification techniques. These include tree-based methods, statistical analysis, and rule-based techniques. We briefly describe in the following the classification models that were used in this evaluation:

**Decision Tree:** C4.5 is one of the most well-known tree-based classifiers. A tree is constructed by using the strategy of divide-and-conquer. C4.5 starts to search an attribute with the highest information gain, and then the data is partitioned into classes according to the attributes' values. Each sub-tree is recursively further partitioned using the same policy until any of the stopping criteria is satisfied; e.g., a leaf node is reached. The decision rules are obtained by traversing the branches of the tree from the root to the leaves. The details of this algorithm can be found in Quinlan (1993).

**Logistic Regression:** Logistic regression (LR) is a type of regression analysis; it has been widely applied in applications of probability classification. The probability of an event may occur is estimated as follows:

$$P(y = 1|x_1, \ldots, x_k) = \frac{e^{\beta_0 + b_1 x_1 + \cdots + b_k x_k}}{1 + e^{\beta_0 + b_1 x_1 + \cdots + b_k x_k}}$$

where $y$ is a binary independent variable representing the occurrence of an event (i.e., $y = 1$ an event occurred and $y = 0$ otherwise), $x_1, x_2, \ldots, x_k$ denote the independent variables and $b_1, b_2, \ldots, b_k$ are the regression coefficients that can be estimated by many methods (e.g., Maximum Likelihood Estimation).

**k-NN:** k-nearest neighbour is one of the most simplest classification techniques. The aim of k-NN is to classify a test instance ($x'$) by finding its neighbourhood, which consists of the $k$ closest instances in the training set. The class label assignment of the test instance is based on the dominance of the classes in the neighbourhood, that is, the test instance should be assigned to the majority class of the $k$ instances. Therefore, this modelling technique has two core elements: the value of $k$, and the distance metric of similarities between instances. It is not easy to set the most proper value of $k$. In this study, we set $k = 3$.

**SVM:** Support Vector Machine (SVM) is an effective and robust technique for classification applications. The SVM main objective is to find the best classification function to separate the data objects.
of the training set into multiple classes. In this study, we are concerned by binary classification application of customer churn prediction, we aim to find the best classification function (i.e., the maximum-margin hyperplane) to partition the training set into two classes. New instances are mapped into the model and are predicted as a class (category) based on the side of the hyper-plane they fall into.

Since the real dataset is not linearly separable, a non-linear classification function \( f(x) \), corresponding to a hyperplane, is used to separate the two classes. A new data instance \( x \) is classified by the function \( f(x) \) based on the following principles:

\[
\begin{align*}
\text{positive.} & \quad \text{if } f(x) < 0 \\
\text{negative.} & \quad \text{if } f(x) > 0
\end{align*}
\]

The classification function can be written as follows:

\[
f(x) = \sum_{i=1}^{n} \alpha_i y_i k(x_i, c) + b
\]

where \( n \) is the number of instances in the training set, \( x_i \) is an instance in the training set, \( c \) is a new instance that does not have a class label, \( k(.,.) \) is a kernel function and \( b \) is a threshold. The parameter \( \alpha_i \) indicates whether \( i \) is a support vector or not, \( <,> \) indicates the inner product of vectors, therefore, the inner product computation for the new instance is only for the “support vectors” rather than the entire training instances. The obtained maximum margin hyperplane is the so called Support Vector Machine. The parameter \( \alpha_i \) can be obtained by solving a convex quadratic programming problem. Burges (1998) proposed that polynomial kernels and Gaussian radial basis functions are often used as kernel functions (Boser, Guyon, & Vapnik, 1992).

**OneR:** OneR is a very efficient rule-based classification learning method. OneR can generate a set of simple rules named “1-rules”, which can classify an object based on a single attribute. To avoid over-fitting, which is a common problem in the domain of classification, OneR requires all intervals to contain more than a user-specified number of examples in the same class. This technique has higher classification efficiency than many other classifiers.

**PART:** PART is an efficient rule-learning classification method; it infers rules by repeatedly building partial C4.5 decision trees; thus PART combines two paradigms for the rule generation; one is for creating rules from decision trees and the other is divide-and-conquer. Once a partial tree has been built, a single rule is extracted from it. The main advantage of PART over the other models lies in its simplicity, and it produces good rule sets that compare favourably with those generated by well-known modelling techniques (e.g., RIPPER).

The telecom data suffers from the imbalance problem since the data consists of a huge number of non-churner records and relatively a very small number of churner records. To overcome this issue we adopted under-sampling technique, which randomly removes some of the majority class samples to reduce its size. Therefore, in our experiment, we train each classifier on a set of samples, which have different churn rate \( \frac{\text{num. churners}}{\text{num. total}} \) in the range [10%, 50%] with the step size of 5%, obtained by applying the under-sampling method to randomly remove the non-churn records. Table 3 shows all the used samples, the third and the fourth columns indicate the number of churners and non-churners in each training sample, respectively; and the last column denotes the size of testing data, which is one fold of the entire dataset.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Churn Rate %</th>
<th>#Train-churn</th>
<th>#Train-nonchurn</th>
<th>#Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>10</td>
<td>4896</td>
<td>44064</td>
<td>20839</td>
</tr>
<tr>
<td>S2</td>
<td>15</td>
<td>4896</td>
<td>27744</td>
<td>20839</td>
</tr>
<tr>
<td>S3</td>
<td>20</td>
<td>4896</td>
<td>19584</td>
<td>20839</td>
</tr>
<tr>
<td>S4</td>
<td>25</td>
<td>4896</td>
<td>14688</td>
<td>20839</td>
</tr>
<tr>
<td>S5</td>
<td>30</td>
<td>4896</td>
<td>11424</td>
<td>20839</td>
</tr>
<tr>
<td>S6</td>
<td>35</td>
<td>4896</td>
<td>9092</td>
<td>20839</td>
</tr>
<tr>
<td>S7</td>
<td>40</td>
<td>4896</td>
<td>7344</td>
<td>20839</td>
</tr>
<tr>
<td>S8</td>
<td>45</td>
<td>4896</td>
<td>5980</td>
<td>20839</td>
</tr>
<tr>
<td>S9</td>
<td>50</td>
<td>4896</td>
<td>4896</td>
<td>20839</td>
</tr>
</tbody>
</table>

4.3.3. Set-up III

In addition to the traditional classification techniques, we also compare the classification performance of the proposed hybrid learning system with several other hybrid classification models. In the following we briefly introduced the main structures of these hybrid models and the necessary parameter setting:

**SePI:** This classification model consists of three main steps: (1) a Main Model is built by applying various single models on training sets; the model that has the best performance is selected as the Main Model; (2) a Discrimination Model is built to found out whether the Main Model performs well on the given application datasets; (3) a Support Model is built by applying various single models on the data that the Main Model does not predict correctly, and the model that has the best performance is selected. Based on these three models, the process of prediction is performed by firstly using the Discrimination Model to find out which model (Main Model or the Support Model) should be used to predict the final results. In this work, the Discrimination Model and the Support Model use NN (Neural Networks), and C5.0 is used for the main model.

**k-NN-LR:** This hybrid classifier combines k-NN and Logistic Regression. This method has two main phases: (1) split the training data \( X \) into \( m \) disjoint data sets, where \( m \) is the number of attributes; (2) transform \( X \) into a new data set \( K(X) \) by employing the k-NN algorithm; then a Logistic Regression classifier is used to train the new data set \( K(X) \). In this study, we set the number of the nearest neighbours to 7.

**KM-BoostedC5.0:** This modelling technique is established by combining clustering techniques and decision tree with boosting. This model is build specifically for a mobile service data set characterized by two types of attributes, which show the information of service usage and revenue contribution, respectively. Therefore, in the first stage, clustering techniques are used to segment the data, and the original data is transformed into data instances with cluster labels only; in the second stage, the obtained new data is used as the input of the boosted decision tree model (C5.0).

In the literature, five clustering techniques (i.e., K-means, K-medoid, SOM, FCM and BIRCH) were applied in the first stage to compare the performance. In this experiment, we use KM (K-means) for segmenting the data set since this literature states that KM generates the most number of models that could beat the benchmark models. In addition, based on attribute information, we segment the real telecom data into clusters by considering two groups of attributes, one group includes the information of demographic profiles and customer account information, the other group includes the information of call details. More information of this hybrid model is shown in Bose and Chen (2009). In order to treat this classification model fairly, in the first stage, instead of setting the number of clusters \( K \) arbitrary, we find the optimal \( K \) values for each data sample from candidate numbers in a range \([1 : 10]\) with step size of 1.
4.4. Results and Discussion

4.4.1. Experiment I

The original k-means clustering aims to minimize the Within-class Sum of Square Error (WSSE). Therefore, we use WSSE to evaluate the clustering performance. We use 9 samples to train the models and all the samples are used in performance evaluation and for comparing the clustering approaches considered in this study. Fig. 2 shows different clustering results produced by the original and weighted k-means. It also shows the clustering performance based on different statistical analysis. In this figure, the horizontal axis represents the churn rate while the vertical axis indicates the objective function (WSSE). Each plot is produced using different data samples. One can see that the weighted method produces better clustering results. In addition, the Figure also shows that the results generated by Path Analysis have lower square error than those of Chi-Square and Information Gain.

In order to demonstrate the performance of the modified clustering method, we display its results. The telecom data has a huge number of attributes, and to better illustrate the function of clustering, we used the PCA (Principal Component Analysis) technique to reduce its dimensionality. We plot the first, the second and the third principal components of all the observations, as illustrated in Fig. 3, X-axes, Y-axes and Z-axes represent the first, second, and third principal component, respectively. Since the customer churn prediction is a binary classification task, we assume that the data is separated into two clusters (cluster_1 and cluster_2). Fig. 3(a) and (b) show the clustering results produced by the weighted k-means and the original k-means, respectively. In the two figures, we show cluster_1 in red and cluster_2 in green. The center of each cluster is also shown, the blue square represents the center of cluster_1, while the black square represents the center of cluster_2. We can clearly see that the left figure has more separable clustering result than the right one, that is, the modified clustering method achieves a better clustering result than the one produced by original method.

In order to prove that this approach is suitable for other applications, we applied it on four other benchmark datasets (Ionosphere, SAheart, vehicle2 and Glass0), which are part of the UCI repository and KEEL (Knowledge Extraction based on Evolutionary Learning) datasets. All the four benchmark datasets require binary classification. Fig. 4 shows 3D clustering results for each dataset. We use the same symbols and colours as in Fig. 3, in order to better show the improvement of the weighted clustering, we display the result of each benchmark data in the most suitable view for observation. The left and right figures represent the modified and the original clustering results, respectively. The left figures separate the datasets more completely than the figures on the right side since the weighted method is able to find the better cluster centres.

4.4.2. Experiment II

Based on the experimental set-up described in Section 4.3.2, we plot the ROC curve for each classification model. Fig. 5(a) compares the ROC curves of the proposed hybrid model and other well-established classification modelling techniques. The horizontal axis and the vertical axis represent the rates of False Churn and True Churn, respectively. We look for the predictive accuracy with high true churn and low false churn.
Each ROC curve is plotted by training the system with different churn rates (one can refer to Table 3 for more details about the samples). From this figure, we can observe that the ROC curve (labelled by NewModKmeans), which is obtained based on the...
Fig. 5. Compare the ROC curves and AUC of different classification techniques in customer churn prediction.

Table 4
An example of the prediction performance of a sample (S1, Churn Rate = 10%) based on different number of clusters. The maximum AUC value in bold.

<table>
<thead>
<tr>
<th>Subset</th>
<th>Performance metric (%)</th>
<th>#Cluster (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2  3  4  5  6  7  8  9  10</td>
</tr>
<tr>
<td>subset1</td>
<td>AUC</td>
<td>84.86 84.98 84.88 85.43 86.71 86.24 87.36 86.78 86.67</td>
</tr>
<tr>
<td>subset2</td>
<td>AUC</td>
<td>87.01 88.71 86.41 88.51 85.12 84.02 86.41 89.02 84.36</td>
</tr>
<tr>
<td>subset3</td>
<td>AUC</td>
<td>87.74 85.27 86.64 86.58 85.76 83.51 86.98 87.83 86.53</td>
</tr>
<tr>
<td>subset4</td>
<td>AUC</td>
<td>87.85 88.16 87.75 86.65 87.83 82.99 87.87 86.15 88.21</td>
</tr>
<tr>
<td>subset5</td>
<td>AUC</td>
<td>86.13 87.2 87.15 85.99 86.02 85.7 86.54 88.27 87.45</td>
</tr>
<tr>
<td>Average</td>
<td>AUC</td>
<td>86.71 ± 1.24 86.86 ± 1.68 86.56 ± 1.07 86.62 ± 1.16 86.28 ± 1.03 85.55 ± 1.91 87.03 ± 0.59 88.01 ± 0.81 86.64 ± 1.44</td>
</tr>
</tbody>
</table>

Table 5
Prediction performance of samples based on different number of clusters. The maximum AUC value in bold.

<table>
<thead>
<tr>
<th>Subset</th>
<th>Performance metric (%)</th>
<th>#Cluster (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2  3  4  5  6  7  8  9  10</td>
</tr>
<tr>
<td>S1</td>
<td>AUC</td>
<td>86.71 ± 1.24 86.86 ± 1.68 86.56 ± 1.07 86.62 ± 1.16 86.28 ± 1.03 85.55 ± 1.91 87.03 ± 0.59 88.01 ± 0.81 86.64 ± 1.44</td>
</tr>
<tr>
<td>S2</td>
<td>AUC</td>
<td>87.8 ± 1.38 87.08 ± 0.82 87.25 ± 0.23 87.66 ± 1.32 87.14 ± 0.6 86.89 ± 0.57 86.57 ± 2.13 87.1 ± 0.77 88.49 ± 1.33</td>
</tr>
<tr>
<td>S3</td>
<td>AUC</td>
<td>87.97 ± 0.73 87.48 ± 0.78 87.92 ± 0.69 87.39 ± 0.58 87.61 ± 1.55 88.15 ± 2.05 87.31 ± 1.27 88.07 ± 0.69 88.35 ± 0.27</td>
</tr>
<tr>
<td>S4</td>
<td>AUC</td>
<td>88.53 ± 1.46 88.54 ± 0.89 88.66 ± 0.71 88.44 ± 0.8 88.91 ± 0.23 88.26 ± 0.69 89.2 ± 0.87 88.55 ± 0.59 88.55 ± 0.38</td>
</tr>
<tr>
<td>S5</td>
<td>AUC</td>
<td>89.02 ± 1.01 88.93 ± 0.92 88.85 ± 0.97 88.6 ± 0.75 88.15 ± 0.85 88.59 ± 0.75 89.01 ± 0.86 89.31 ± 0.97 89.25 ± 0.91</td>
</tr>
<tr>
<td>S6</td>
<td>AUC</td>
<td>88.99 ± 1.39 89.13 ± 1.02 89.46 ± 1.04 89.61 ± 1.04 89.38 ± 1.04 89.21 ± 0.74 89.01 ± 0.51 88.75 ± 0.9 88.73 ± 0.81</td>
</tr>
<tr>
<td>S7</td>
<td>AUC</td>
<td>89.44 ± 0.79 89.15 ± 0.75 89.51 ± 0.65 88.88 ± 1.69 89.39 ± 1.26 88.7 ± 1.17 88.65 ± 1.36 88.03 ± 0.85 88.07 ± 1.06</td>
</tr>
<tr>
<td>S8</td>
<td>AUC</td>
<td>89.32 ± 1.3 88.51 ± 1.02 89.21 ± 0.64 89.03 ± 0.58 88.8 ± 1.37 88.94 ± 1.09 88.97 ± 0.91 89.21 ± 1.2 88.81 ± 1.66</td>
</tr>
<tr>
<td>S9</td>
<td>AUC</td>
<td>87.54 ± 1.46 89.4 ± 0.71 88.36 ± 1.43 88.32 ± 0.64 88.32 ± 0.93 88.36 ± 0.89 88.43 ± 0.89 87.91 ± 0.63 88.46 ± 0.99</td>
</tr>
</tbody>
</table>

Table 6
Compare the Overall Accuracy (OA) of the hybrid-model with the traditional classifiers based on all samples and Wilcoxon signed-rank test statistic values for testing the statistical significance when comparing the hybrid-model with the other classifiers. The maximum accuracy values in bold.

<table>
<thead>
<tr>
<th>Sample data</th>
<th>Performance metric (%)</th>
<th>Hybrid-Model</th>
<th>Tree (C4.5)</th>
<th>Logistic</th>
<th>knn</th>
<th>SVM</th>
<th>PART</th>
<th>OneR</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>OA</td>
<td>96.75</td>
<td>97.53</td>
<td>97.54</td>
<td>94.31</td>
<td>96.68</td>
<td>96.57</td>
<td>96.61</td>
</tr>
<tr>
<td>S2</td>
<td>OA</td>
<td>97.51</td>
<td>96.79</td>
<td>97.03</td>
<td>92.94</td>
<td>96.39</td>
<td>95.52</td>
<td>96.52</td>
</tr>
<tr>
<td>S3</td>
<td>OA</td>
<td>97.09</td>
<td>96.11</td>
<td>96.4</td>
<td>91.46</td>
<td>95.67</td>
<td>94.5</td>
<td>93.43</td>
</tr>
<tr>
<td>S4</td>
<td>OA</td>
<td>96.67</td>
<td>95.32</td>
<td>95.56</td>
<td>89.89</td>
<td>95.17</td>
<td>93.618</td>
<td>92.72</td>
</tr>
<tr>
<td>S5</td>
<td>OA</td>
<td>95.13</td>
<td>94.46</td>
<td>94.65</td>
<td>88.27</td>
<td>95.03</td>
<td>92.68</td>
<td>92.65</td>
</tr>
<tr>
<td>S6</td>
<td>OA</td>
<td>93.89</td>
<td>93.72</td>
<td>93.89</td>
<td>86.63</td>
<td>95.01</td>
<td>91.35</td>
<td>92.66</td>
</tr>
<tr>
<td>S7</td>
<td>OA</td>
<td>94.61</td>
<td>92.41</td>
<td>92.35</td>
<td>84.84</td>
<td>94.4</td>
<td>90.04</td>
<td>92.67</td>
</tr>
<tr>
<td>S8</td>
<td>OA</td>
<td>94.44</td>
<td>91.46</td>
<td>91.31</td>
<td>82.98</td>
<td>87.63</td>
<td>88.87</td>
<td>76.74</td>
</tr>
<tr>
<td>S9</td>
<td>OA</td>
<td>90.01</td>
<td>90.11</td>
<td>90.18</td>
<td>81.04</td>
<td>78.51</td>
<td>87.45</td>
<td>76.74</td>
</tr>
</tbody>
</table>

Wilcoxon Statistic Value (W) W+=44, W=1 W+=44, W=1 W+=45, W=0 W+=42, W=3 W+=45, W=0 W+=45, W=0
proposed hybrid model, is close to the left-upper corner most. Therefore, it compares favourably with the other six different classification models. We also plotted the ROC curve (labelled by NewKmeans) for the hybrid model by applying the original k-means. The figure illustrates that it has similar accuracy with Decision Tree for most of the points (with either higher rate of True Churn or lower rate of False Churn) and outperforms many other classifiers, but does not perform as good as the one based on weighted k-means. Fig. 5(b) compares the hybrid model-based learning system with the six classifiers by using Area Under ROC Curve as the evaluation metric. In this figure, the horizontal and vertical axes represent the churn rate and the AUC values, respectively. Each curve is plotted by considering all samples, and we can obviously see that the AUC values obtained by the hybrid system are all higher than the ones obtained by other classification models no matter what the churn rates of the samples are.

In addition, the original k-means, decision tree, logistic regression, PART, SVM, KNN, and OneR are successively followed by the proposed model. For several samples, the hybrid system achieves nearly two percentage points higher on AUC than decision tree. The k-means based hybrid model, in fact, has similar performance than decision tree, it achieves a little higher AUC only when the churn rate becomes relatively high (i.e., 40%, 45%, 50%).

In terms of the number of clusters (K), as mentioned in Section 4.3.2, each sample might have different optimal K due to different churn rates and data objects distributions. As mentioned in previous section, we applied the cross validation method for all samples. Table 4 shows an example of obtaining the final prediction performance and deciding the optimal K for a training sample (here S1 is an example sample), and AUC is considered as the a metric measure. The last row indicates the average performance and the standard deviation of the 5 folds of validation, and the maximum value is obtained for K = 9. We did the same cross validation on each training samples. Table 5 shows the prediction performance based on all samples, and the maximum AUC values acquired by the optimal K are marked in bold. Moreover, it is good to see that the standard deviation for all the obtained accuracy metric values is very small, especially the acquired optimal performance metric in many cases. This reflects the consistency and stability of the proposed model.

ROC and AUC can objectively evaluate the classification accuracy by considering the ratios of both true and false positive. In a sense, they can better examine the skills of models than many other evaluation methods; therefore they are commonly used as a metric for the classifiers. Moreover, the overall accuracy (OA) is also important for evaluating a prediction model comprehensively.

Table 6 compares the proposed hybrid model with the traditional classifiers on the overall accuracy rate. In this table, each row indicates the accuracy rate obtained from each sample, and we highlight the maximum accuracy rates in bold. Each OA of the hybrid-model is calculated for the corresponding optimal K (number of clusters). It is interesting to see that the new model compares favourably with the other ones in most of the cases; in addition, SVM achieves the best accuracy on S6, decision tree and the proposed one has more or less the same accuracy on S9.

Furthermore, in order to study the significance of this comparative study, which decides whether there is a significant difference between the proposed model and the traditional classifiers, it is necessary to use a statistical test method. We utilise the Wilcoxon signed rank test on the overall accuracy rates since it is a non-parametric hypothesis test and does not require the data to be normally distributed. We simply point out the data and parameters related to the test rather than describe the Wilcoxon test steps in detail. In our case, the paired data would be the third column in Table 6 and one of the remaining columns (i.e., any column indicating the accuracy rates of a traditional classifier), the paired data size N = 9. We take the significance test between hybrid-model and SVM as an example. We firstly propose the null hypothesis (i.e., there is no significant difference between the hybrid-model and SVM), after ranking the absolute differences of the paired data, the positive total rank W+ = 42, while the negative total rank W− = 3. Thus, the Wilcoxon test statistic W = 3. With a data sample size N = 9, the critical value for a two-tailed test at α = 5% and one-tailed test at α = 2.5% is 6. The test statistic value is 3 (less than 6), therefore, we can reject the null hypothesis and conclude that there is a significant difference between the hybrid-model and SVM with 95% confidence, furthermore, it is likely that the hybrid-model is better than SVM with 97.5% confidence. To test the significant difference between the hybrid-model and other classifiers, the positive and negative total rank values are shown in Table 6, each with smaller total rank equal to the Wilcoxon test statistic W. Thus, we can conclude that the prediction accuracy of the hybrid-model is significantly better than the other six classifiers.

4.4.3 Experiment III

Fig. 6 compares the prediction performance between the proposed hybrid model and other hybrid classifiers described in Section 4.3.3. The figure illustrates that the proposed hybrid model obviously compares favourably well with the other three hybrid classification methods. NewKmeans (the new hybrid model based on original k-means) outperforms k-NN-LR and KM-Boosted C5.0, in addition, this figure shows that NewKmeans is as good as SePl in the first several points, which are the results of the samples having relative lower churn rates; for the samples with higher churn rate, NewKmeans produces lower ratio of False Churn than SePl.

In addition to the True Churn and False Churn illustrated in the ROC Curve, we also compare our model to the above mentioned hybrid classification methods. To do so, we use AUC and the overall accuracy and the results are shown in Table 7. In this table, each row depicts the performance (AUC and OA) achieved by each hybrid technique and the maximum AUC and accuracy are highlighted in bold. As one can notice, our proposed hybrid model shows the absolute advantages on both AUC and OA for all the samples used (S1, ..., S9).

In order to show that the proposed hybrid modelling technique is more general and can be applied to many other classification or prediction applications, we applied this system on 22 benchmark datasets, which are collected from the UCI machine learning repository and KEEL datasets. Table 8 shows the AUC values of all the used datasets, which are calculated by the hybrid model-based
over, most of well-established classification techniques are based on a single model. This paper presents a hybrid model-based classification learning system, which integrates a weighted k-means clustering and a classic inductive rule learning method (FOIL). The proposed technique combines the clustering and classification data mining approaches.

To evaluate the prediction accuracy of the proposed approach, we use ROC and AUC as the measure metric, and compare it with several well-known classifiers. The experimental results show that the proposed system compares favourably well with these well-known classifiers. In addition to the telecom data, we applied our techniques on 22 benchmark datasets, and compare it with other classifiers. The results show that a large number of the benchmark data gain the maximum accuracy when applying the hybrid model.

For future work, we will be committed to improving the proposed learning system by considering several issues. Firstly, as the pre-processing in data mining, outliers and redundant data examples can be detected, and for this hybrid model, eliminating outliers would greatly contribute to a better clustering result. Secondly, other clustering algorithms, which can decide about the number of clusters, might be applied in the first stage. Finally, other clustering algorithms, which can decide about the number of clusters, might be applied in the first stage.
Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.eswa.2013.04.020.

References


