Chapter 6

Performance Evaluation of CHURN

We evaluated our approach on 19 different data sets from the UCI data collection as well as different data sets for forecasting the behaviour of an optimisation heuristic within a hyper heuristic framework. Tenfold cross-validation was used to derive the classifiers and error rates in the experiments. Cross validation is a standard evaluation measure for calculating the error rate on data in machine learning. Three popular classification techniques have been compared to MMAC in terms of classification accuracy (PART, RIPPER, CBA) in order to evaluate the predictive power of the proposed method. The choice of such learning methods is based on the different strategies they use to generate the rules. Since CBA, PART and RIPPER are suitable for traditional classification problems, the classification accuracy derived by only the top-label evaluation measure has been used for a fair comparison.

All experiments were conducted on Pentium IV 1.6 GHz PC under Windows XP. The experiments for PART and RIPPER were conducted using the Weka software system. Weka stands for Waikato Environment for Knowledge Analysis, which is an open java source code for the machine-teaching community that includes implementations of several data-mining tasks such as classification, clustering, association rule mining and regression. CBA experiments were conducted using a VC++ implementation version provided in CBA, and finally MMAC was implemented using Java.

Many studies have shown that the support threshold plays a major role in the Overall classification accuracy of the set of rules produced by existing associative classification techniques [17, 6]. Moreover, the support value has a large impact on the number of rules produced in the classifier as well as the processing time and storage needed during rules discovery. From our experiments, we noticed that support rates ranging between 0.02 to 0.05 usually achieve the best balance
between accuracy rates and the size of the resultant classifiers. Moreover, the classifiers derived when the support was set to 2 and 3 achieved high accuracy and are often better than that of decision trees rule (PART), RIPPER and CBA. Due to this, the MinSupp value was set to 0.03 in the experiments. The confidence threshold, on the other hand, is less complex and unlike the support value does not have a large effect on the behaviour of associative classification methods, and therefore it was set to 0.30.

6.1 Binary and traditional data results

We have evaluated 19 selected data sets from, six of which were reduced by ignoring their integer and/or real attributes. Several tests using tenfold cross validation have been performed to ensure that the removal of any real/integer attributes from these data sets does not significantly affect the classification accuracy. As a result, we only considered data sets where the error rate was not more than six times worse than the error rate obtained on the same data set before the removal of any real/integer attributes. Table 1 represents the classification accuracy of the classifiers extracted by PART, RIPPER, CBA and MMAC on the 19 benchmark problems. The accuracy of MMAC has been derived using the top-label evaluation measure. Our algorithm outperforms the rule-learning methods in terms of accuracy rate, and the won-loss-tied records of MMAC against PART, RIPPER and CBA are 13–6–0, 15–4–0 and 15–4–0, respectively. The results shown in Table 1 indicate that our proposed algorithm outperforms CBA in terms of error rate. One of the principle reasons for this appears to be that MMAC often generates a few more rules than CBA. The increase in accuracy suggests that this is not simply over fitting and would likely justify the small increase in classification rate for MMAC over CBA in applications. However, in some cases, such as the "CRX" data set, the number of rules is large, even though every rule represents at least one training object. Thus, a post-pruning method, such as pessimistic error pruning, may be useful in such cases. Figure 1 shows the runtime of MMAC, CBA, PART and RIPPER for 19 UCI data sets. It indicates that the associative-based techniques MMAC and CBA are slower than the traditional
techniques PART and RIPPER. Particularly, since CBA adapts the a priority approach, which is a resource- and time-consuming approach, making it the slowest algorithm for building classifiers. Furthermore, PART is the fastest algorithm to construct a classification system due to its simple strategy to discover rules. The figures also suggest that our proposed algorithm is typically slower than traditional classification techniques such as RIPPER and PART. However, they also indicate that MMAC compares well in term of processing time with the CBA algorithm.

![Graph showing the difference in accuracy between MMAC evaluation measures and CBA algorithm](image)

Fig. 1 Difference in accuracy between MMAC evaluation measures and CBA algorithm
Table 1 Classification accuracy of PART, RIPPER, CBA, and MMAC.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>PART</th>
<th>RIPPER</th>
<th>CBA</th>
<th>MMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (since how long he is the customer)</td>
<td>96.3</td>
<td>73.3</td>
<td>81.5</td>
<td>76.7</td>
</tr>
<tr>
<td>Change in price plan</td>
<td>54.5</td>
<td>88.6</td>
<td>96.5</td>
<td>84.8</td>
</tr>
<tr>
<td>Air charge</td>
<td>88.6</td>
<td>95.6</td>
<td>89.6</td>
<td>77.5</td>
</tr>
<tr>
<td>MTNL charge</td>
<td>77.6</td>
<td>90.7</td>
<td>79.8</td>
<td>69.0</td>
</tr>
<tr>
<td>Total call duration</td>
<td>98.6</td>
<td>96.6</td>
<td>93.4</td>
<td>94.3</td>
</tr>
<tr>
<td>Total charges</td>
<td>76.8</td>
<td>88.6</td>
<td>89.6</td>
<td>96.6</td>
</tr>
<tr>
<td>No. of incoming calls</td>
<td>65.8</td>
<td>97.7</td>
<td>99.0</td>
<td>56.9</td>
</tr>
<tr>
<td>No. of outgoing calls</td>
<td>95.6</td>
<td>89.6</td>
<td>77.5</td>
<td>98.9</td>
</tr>
<tr>
<td>Distinct numbers called</td>
<td>89.8</td>
<td>66.8</td>
<td>98.7</td>
<td>94.5</td>
</tr>
<tr>
<td>Status (continuing/left)</td>
<td>78.7</td>
<td>98.9</td>
<td>99.8</td>
<td>87.8</td>
</tr>
</tbody>
</table>

6.2 Multi-label scheduling data results

Details of several solution runs generated by a hybrid hyper heuristic, named Peckish, for solving a complex scheduling problem are provided by Thabtah et al. Cowling and Chakhlevitch. The hyper heuristic is a robust, general approach that tends to find good solutions, rather than optimal ones for large and complex scheduling and optimisation problems. One can consider a hyper heuristic approach as a supervisor that manages the choice of which low-level heuristic to select from the available ones for the problem during the process of building a schedule. The Peckish hyper heuristic often selects the low-level heuristic that leads to the maximum improvement to the objective function (if one exists). If none of the existing low-level heuristic improves the objective function, the Peckish hyper heuristic selects one randomly. A low-level heuristic is a simple rule or method that yields a small change in the schedule. Often these low-level heuristics are based on human methods of constructing the schedule, such as adding an event, deleting an event or swapping two events. The Peckish hyperheuristic often selects the low-level heuristic that leads to the maximum improvement to the objective function (if one exists). If none of the existing low-level heuristic improves the objective function, the Peckish hyper heuristic selects one randomly. A low-level heuristic is a simple rule or method that yields a small change in the schedule. Often these low-level heuristics are based on human methods of constructing the schedule, such as adding an event, deleting an event or swapping two events.
We have used nine solution runs, in which each solution contains 500 iterations of applied low-level heuristics. In addition, ten different low-level heuristics (LLH 1, LLH 2, LLH 20, LLH 27, LLH 37, LLH 43, LLH 47, LLH 58, LLH 68 and LLH 74) have been used to generate each solution. Each solution consists of six different attributes and a number of instances that maximise the objective function of the optimisation problem. Table 5 indicates the features of the data sets used in the experiments. Table 6 represents part of a solution run generated by the Peckish hyper heuristic where columns \textit{LLH}2 and \textit{LLH}1 represent the low-level heuristics applied at the previous two iterations. Column \textit{LLH} represents the current low-level heuristic that improved the objective function and column \textit{Imp} represents the improvement on the objective function value. Finally, the column \textit{Apply} represents whether or not the selected low-level heuristic has been applied by the hyper heuristic. The data generated by the hyper heuristic is multi-label, since at each iteration there could be more than one low-level heuristic.
that improves the objective function. For example, at the first iteration in Table 6, there are three low-level heuristics (LLH 2, LLH 43 and LLH 4) that improve the objective function. Thus, there are three class labels associated with the instance (1, 1). Generally, each training instance in the scheduling solutions may associate with at least; one class label. The evaluation measures presented with MMAC have been compared on the nine solution runs produced by the Peckish hyperheuristic, with regard to accuracy, and rules features. Figures 2 and 3 represent the relative prediction accuracy and indicate the difference in the classification accuracy of the MMAC evaluation measures with respect to those derived by CBA and PART. The relative prediction accuracy numbers shown are calculated using the formula 

\[
\frac{\text{Accuracy}_{\text{MMAC}} - \text{Accuracy}_{\text{PART}}}{\text{Accuracy}_{\text{PART}}} \quad \text{for PART and}
\]

\[
\frac{\text{Accuracy}_{\text{MMAC}} - \text{Accuracy}_{\text{CBA}}}{\text{Accuracy}_{\text{CBA}}} \quad \text{for CBA.}
\]

After analysing the charts, we found that there is a consistency between the top-label and label-weight measures, since in most cases both of them consider only one class in the prediction. The top-label takes into account the top-ranked class, and the label-weight considers only the weight for the predicted class that matches the test case. Thus, both of these evaluation measures are applicable to traditional single-class classification problems. On the other hand, the any-label measure considers any class in the set of the predicted classes as a hit whenever it matches the predicted class, regardless of its weight or rank. Furthermore, the support-weight measure balance between any-label and top-label measures because it considers the top class the fittest on one hand and assigns each of the rest of classes a weight on the other hand. It should be noted that the relative accuracy of MMAC evaluation methods against data set number 8 in Figs. 2 and 3, is negative since CBA and PART achieved a higher classification rate on this particular data set.

An investigation on the rules features derived from the scheduling data runs has been carried out. Figure 4 shows the number of rules extracted from each data
set, categorised by the number of class labels. Unlike most associative techniques, our proposed algorithm is able to extract rules that are associated with up to four labels. This is one of the principle reasons for improving the classification accuracy within applications. Figure 4 also demonstrates that the majority of the rules created from each scheduling run are associated with one or two labels. It turns out that this is due to the fact that, during each iteration, often only one or two low-level heuristics improve upon the objective function in the scheduling problem. Thus, each training instance often corresponds to just one or two class labels. Figure 5 shows the distribution of the training instance association, with class labels for one scheduling data set where instances that are correlated with one class label dominate the rest.
Finally, analysis of the rules sets also indicates that MMAC derives more rules than PART and CBA for the majority of the data sets. One principle reason for extracting more rules is due to the recursive learning phase that MMAC employs. This discovers hidden information that most of the associative classification techniques discard, as they only extract the highest confidence rule for each frequent item.
6.3 Conclusions

A new approach for multi-class multi-label classification rules has been proposed that has many distinguishing features over traditional and associative classification methods: (1) It produces classifiers that contain rules with multiple labels, (2) It presents four evaluation measures for determining accuracy that are applicable to a wide range of applications, (3) It employs an efficient method for discovering rules that requires only one scan over the training data, (4) It employs a detailed ranking method, which prunes redundant rules, and ensures only effective ones are used for classification. Performance studies on 19 data sets from UCI data collection and nine hyper heuristic scheduling runs indicate that our proposed approach is effective, consistent and has a higher classification rate than PART, CBA and RIPPER algorithms. In addition, the proposed algorithm is able to extract rules with up to four multiple labels from the scheduling data sets, which results in a higher classification accuracy for test instances. In further work, we anticipate extending the method to treat continuous data and create a hyper heuristic approach that learns “on-the-fly” which low-level heuristic method is the most effective.
Chapter -7

Scope of Future Work

Our project work illustrates how an insight into the process of “Customer Churn” can be gained using the Analytical CRM techniques. The result of the churn analysis can be refined by taking into consideration some more parameters pertaining to Customer Satisfaction & Brand Building. Various other parameters which also affect the churn but cannot be modeled directly into numeric value can be taken into account for future study. Various inputs in call centre or customer care centre are entered in Text Format.

Some such parameters are Customer Complaints, Dealer Recognition, Size of Dedicated Office Space; Brand Building focus of Network Provider (through advertisements & other marketing investments) can also be taken into consideration for more accurate churn results. The input to such parameters has to be converted from Text Format to Numeric Format for Data Analysis & Data Mining. Such conversion can be done by using Text Mining Software. Text Mining Software’s convert Text Data to Numeric Data using various Numeric Mapping models. Some of the popular Text Mining Software which can be used are IBM Product (Text Miner) & another Text Miner from Poly analyst Corporation.

Some of the numeric parameters as well which we did not take into account in this research, but which will enhance prediction accuracy of Churn are related to customer ease, better pricing & enhanced network connectivity of the service provider. These numeric nodes can also be added into decision tree construction for better accuracy of churn prediction results.

More the network has facilities of enlarged connectivity & better rate plans; less will be the Customer Churn. We in our study did not study parameters pertaining to International connectivity such as International Roaming facility, call rates, existing
broad tie-ups with various other networks on global arena for providing smooth connectivity worldwide to the roaming customers. Taking these parameters as well in our Churn Analysis will give better visibility into reasons of customers leaving the network & subsequently controlling them.

In future study, we can get more accurate results if we take into consideration some more parameters pertaining to Billing, Network & Activation related data. Future work can be performed by taking into consideration all or some of the above mentioned additional parameters. This will surely provide better Reality check on Telecom Post Paid and Prepaid Network users churn analysis. We can use the same Data mining algorithm for finding nodes/parameters which play fundamental role in customer churning. We can then find the more effective parameters out of all comprehensive list of churn reasons possible. Controlling the Churn would surely become much more easier once the reasons of Churn are nailed down to more deeper level and subsequent steps taken by Marketing, Sales & Strategy departments to develop various rate plans, incentives, international tie-ups, brand visibility, better service to customer complaints, & such other steps.

Such steps will enhance customer satisfaction level with existing network provider, & will result in reducing the churn to minimum percentage. Similar Churn analysis & research can also be carried on Industry verticals other than Telecom as well, which are more susceptible to customer leaving the system at faster pace, & controlling their exit from the system.

A new classification approach, called CPAR, is developed to integrate classification and association rule mining. Based on our performance study, CPAR achieves high accuracy and efficiency, which can be credited to the following distinguished features: (1) it uses greedy approach in rule generation, which is much more efficient than generating all candidate rules, (2) it uses a dynamic programming approach to avoid repeated calculation in rule generation, (3) it selects multiple literals and builds multiple rules simultaneously, and (4) it uses expected accuracy to
evaluate rules, and uses the best $k$ rules in prediction. CPAR represents a new approach towards efficient and high quality classification. It is interesting to further enhance the efficiency and scalability of this approach and compare it with other well-established classification schemes. Moreover, the strength of the derived predictive rules also motivates us to perform an in-depth study on alternative approaches towards effective association rule mining.

A new approach for multi-class, and multi-label classification has been proposed that has many distinguishing features over traditional and associative classification methods in that it (1) produces classifiers that contain rules with multiple labels, (2) presents three evaluation measures for evaluating accuracy rate, (3) employs a new method of discovering the rules that require only one scan over the training data, (4) introduces a ranking technique which prunes redundant rules, and ensures only high effective ones are used for classification, and (5) integrates frequent items set discovery and rules generation in one phase to conserve less storage and runtime.

Performance studies on 19 datasets from Weka data collection and 9 hyper heuristic scheduling runs indicated that our proposed approach is effective, consistent and has a higher classification rate than the-state-of-the-art decision tree rule (PART), CBA and RIPPER algorithms. In further work, we anticipate extending the method to treat continuous data and creating a hyper heuristic approach to learn “on the fly” which low-level heuristic method is the most effective.