Applying Reinforcement Learning for Customer Churn Prediction

Applying Reinforcement Learning for Customer Churn Prediction

M Panjasuchat and Y Limpiyakorn

Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand

Yachai.L@chula.ac.th

Abstract. Customer churn prediction is one of the biggest challenges for companies nowadays. Since the loss of customers directly affects the organization's reputation, financial and growth plans. Customer behaviours may change due to any uncontrollable factors or unexpected circumstances, resulting in changing patterns of data. This may aggravate the prediction performance of the classifiers generated from supervised learning technique considered as Passive learning. This paper has thus proposed applying the technique of reinforcement learning for customer churn prediction. The model of Deep Q Network (DQN) is implemented and adapted for learning on the selected customer churn dataset used for classification tasks. We simulate a different distribution dataset with shuffle sampling to embrace pattern changes. The performance of the selected classifiers, compared to DQN, has been evaluated with four measures: accuracy, precision, recall, and F1. The results showed that as Active learner, DQN outperformed the selected classifiers, XGBoost, Random forest, and kNN, when data have been enlarged and evolving with emerging new patterns.

1. Introduction

The number of customers is an important factor contributing to business success. They are the persons who generate revenues for the companies. Research finding [1] reported that retaining an existing customer costs 5 times lower than acquiring a new customer. And more than half of company's revenues come from existing customers. Besides it may take some time for new customers to become loyal as the old customers. Therefore, companies, especially in the telecommunication field, have to maintain or decrease the number of customer churn, who stop using the companies' services or move to other competitors. Churn prediction plays an important role to help detecting customers who are potentially to leave the company, so that companies can provide personalized services to induce them to stay with the company. In recent years, Machine Learning, a discipline of AI, is a technique widely used as the effective solution for several domain problems. A lot of research used classification models generated from supervised learning for churn analysis and prediction. Rai et al. [2] proposed a classification model using decision tree, to analyze the reasons of customer churn, and to predict whether the loss of customers. The result showed that the decision tree classifier outperformed the logistic regression model. Bhatnagar et al. [3] studied the k-Nearest Neighbor (kNN) model for churn prediction in a telecommunication company. The findings reported the strength and weakness of kNN. For the strength, kNN is good for nonlinear, easy to understand, high accuracy, and able to apply for both classification and regression. Still, computation is expensive, i.e. taking plenty of time when predicting. The result showed that kNN yielded 2% higher accuracy than logistic regression. Besides,

researchers have focused on ensemble-based classification techniques. These techniques provide higher accuracy than single model. Keramati et al. [4] studied some well-known algorithms including decision tree, artificial neural networks, k-nearest neighbors, and support vector machines. Each algorithm has its own specialties. The authors proposed an approach to hybridizing these techniques for improving the model prediction. The hybrid methodology achieved the best performance, above 95% accuracy for recall and precision. One of the challenges in customer churn prediction is patterns of data periodically changed. The circumstance may aggravate the performance of the classifiers generated by supervised learning as the model generalization is based on the training dataset. On the contrary, Reinforcement Learning (RL) is considered an active learning able to adapt to different distributions of data. That is, the RL Agent has the ability of self-learning and improving the performance by interacting with environments. Lopez-Martin et al. [5] applied RL for supervised problems in the network security domain. The research showed that RL can improve prediction performance and classify the intrusion faster compared to the current machine learning techniques such as SVM, kNN, Naïve Bayes, etc. The experiments were conducted on two datasets: unbalanced dataset and different distribution of class labels in training and test sets. The results showed that RL could adapt to the data with different distribution from the training data. This research, therefore, aims at studying the model performance for predicting customer churn in telecommunication domain where the dataset is growing with pattern changes. Compared to the classifiers generated from supervised learning, considered as passive learning, the prediction using reinforcement learning model, or active learner, is evaluated by common measures used in literature.

2. Background

2.1 Reinforcement learning (RL)

RL is considered Active learning where training dataset is not required. During the training process, an agent learns behaviours through trial-and-error interactions with an environment. The goal of the agent is to find the set of actions that can maximize the total rewards. Normally, the problems to be solved in RL are modelled as a Markov Decision Process (MDP). An MDP consists of four elements that are a set of states (S), a set of actions (A), probability of transition (T) from every state-action pairs to every possible new state, and reward function of every transition from current state to next state with an action (R). In an MDP, the probability of transition (T) follows the Markov property which means that the next state depends on the current state and action. As shown in figure 1, at each time t, an agent discovers the current state (S_t) and takes an action (A_t). Then, environment will give reward (R_t) which can be positive or negative. Actions change the environment and can lead to a new state (S_{t+1}) where the agent can perform another action and keeps on learning until the end of episode. The goal of RL is to learn sequences of actions that will lead to maximum long-term expected reward. RL commonly consists of three elements. First, *policy* is the mapping of optimal actions in any state. Second, *reward signal* returns the long-term rewards used in defining the good or bad events for the agent. Third, *value function* consists of state-value function (V-function) and action-value function (Q-function). The V-function is the expected return when starting in state s and following policy π. The Q-function is the expected return when starting in state *s*, taking action *a*, and thereafter following policy π.

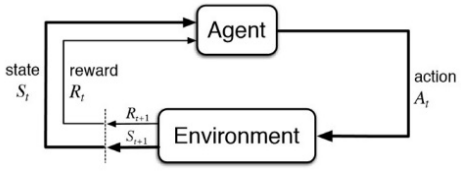

Figure 1. Structure of reinforcement learning [6].

2.2 Q-learning

Q-learning [6] is one of the basic and popular algorithms to get started with reinforcement learning. It is regarded as the value-based learning algorithm where the agent first approximates the value function,

or Q-value of each state prior to make a decision with the policy strategy. In value based RL, the policy is usually short and simple such as greedy or epsilon greedy policy. When predicting Q-value, it commonly uses: 1) tabular representation of Q-value known as Q-table, or 2) universal function approximation known as neural network. The latter is much more flexible and widely used in more complex simulation. During learning, Q-value will be iteratively updated using temporal difference method where the value of one state is bootstrapped based on the value sequel or prequel of the known state. As in equation (1), Q-value is updated using Bellman equation by taking two inputs, state and action, from sampled experience in form of states, actions, rewards, and next states.

$$Q(s,a) = r(s,a) + \gamma \max_{a} Q(s',a) \tag{1}$$

3. Dataset

A telecommunication dataset from Kaggle (https://www.kaggle.com/abhinav89/telecom-customer) is selected. The binary-class dataset contains 100,000 rows and 99 attributes (78 numeric data types, 21 categorical data types). All categorical attribute values are transformed to numeric by one-hot encoding. The class Churn is labelled 1, and Not Churn labelled 0. The data pre-processing is performed including:

- Remove all the samples containing missing values, resulting in the dataset size of 78,334 rows;
- Remove KEY attribute;
- After the previous two steps, random shuffle the dataset to create another dataset, called dataset2 containing the same number of samples;
- Train the random forest models to obtain the set of 21 relevant attributes.

The dataset details are described in table 1. Dataset3 is combination of dataset1 and dataset2.

Table 1. Proportion of label by training and test set of both dataset1 and dataset2

	Churn			Not churn			Total	
	Train	Test	Total	Train	Test	Total	Train	Test
Dataset 1	30,379	7,595	37,974	32,288	8,072	40,360	62,667	15,667
Dataset 2(Shuffle)	30,379	7,595	37,974	32,288	8,072	40,360	62,667	15,667
Dataset 3	60,758	15,190	75,948	64,576	16,144	80,720	125,334	31,334

4. Methodology

The RL technique selected in this work is Deep Q-Network (DQN). In order to apply reinforcement learning for supervised problems, the modified DQN model (figure 2) is presented including:

- mapping the attribute vector or features of data to states, while class labels become actions;
- the predicted label is compared with class labels in the customized environment to generate rewards;
- to satisfy the goal of maximizing the prediction score, we shaped the value of rewards to +20 for correct prediction, otherwise setting to -10 if missed. The reward is used for model optimization;
- the structure of neural network consists of 4 fully connected layers with 2 hidden layers, each of which contains 256 neurons. ReLU activation is used for all layers except output layer uses linear activation. The learning rate is 0.001 and Adam is optimizer algorithm.

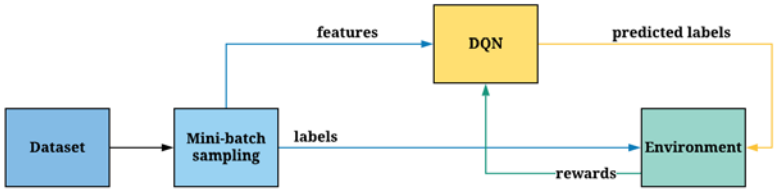

Figure 2. Modified DQN to learn supervised learning dataset.

When applying reinforcement learning on supervised learning data with class labels, there will be no sequential transition between states as shown in figure 4, compared to normal state transition in figure 3. As a result, it is not necessary to calculate the value function from the next state, and the target Q-value of states will be set to the value of reward given that state and action as shown in equation 2. The algorithm then optimizes the predicted Q value toward the target.

$$Q(s,a) = r(s,a) \tag{2}$$

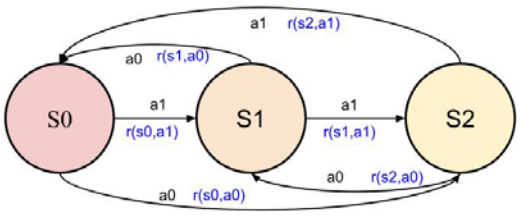

Figure 3. Normal state transition.

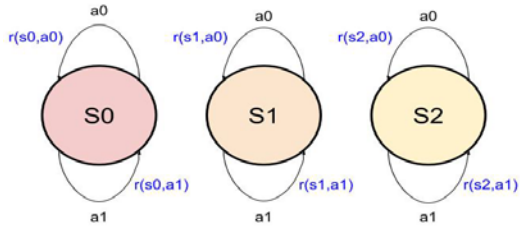

Figure 4. State transition with supervised learning dataset.

5. Experiments and evaluation

The model, DQN, was trained on batch data containing random samples of 20,000 instances. It was trained for 200 episodes. It is observed that the accuracy of the model increases rapidly during the early episodes and then still grows steady, showing the ability of fast adaptive within a few episodes learning. Figure 5 shows the accuracy of test set of dataset1. Since reinforcement learning is active learning able to continuously learn, the model performance on dataset2 created with shuffle sampling to put in pattern data changes still shows well adaptability, as reported in figure 6.

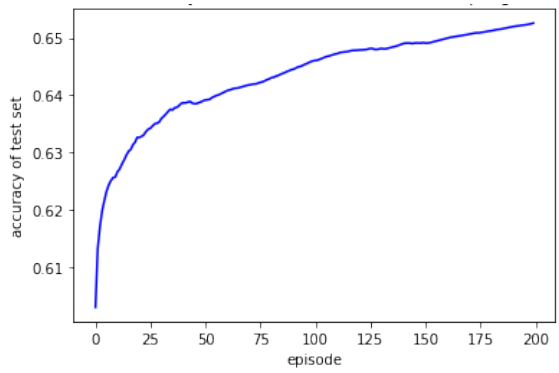

Figure 5. Accuracy of test set of dataset1.

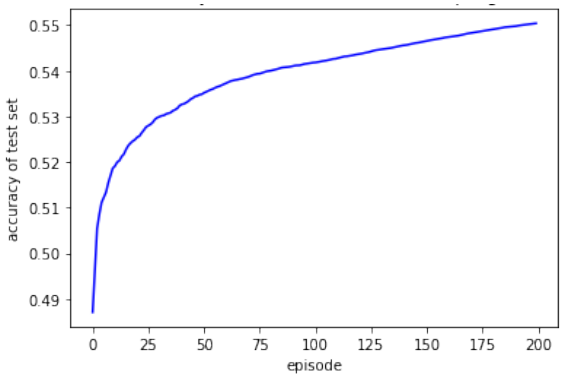

Figure 6. Accuracy of test set on dataset2.

For the selected classifiers, Random forest, XGBoost, kNN, the models were trained twice: 1) using dataset1, and 2) using dataset3 created by merging dataset1 with dataset2 (shuffled samples) to

simulate the scenario that the new supervised models need to reconstructed due to data pattern changes that may cause the performance drop. After removing the rows containing missing values, the size of dataset1 is 78,334 which is then separate into training and test sets with the ratio 80:20, or containing 62,667 and 15,667 instances, respectively. Regarding kNN, k is set to 23 obtained from train-validate-test stopping. The model performance is evaluated by four measures: 1) accuracy, 2) precision, 3) recall, and 4) F1, as reported in table 2. We found that the performances of all models drop, when the dataset contains pattern changes. Nevertheless, the DQN outperforms compared to the others.

Table 2. Summary of performance of the four models.

Measure	Dataset1				Dataset containing pattern changes			
	DQN	XG Boost	Random forest	kNN	DQN	XG Boost	Random forest	kNN
Accuracy	**65.26%**	61.47%	60.14%	55.44%	**55.04%**	55.50%	50.99%	51.44%
Precision	**63.71%**	60.39%	59.21%	54.47%	**52.65%**	54.73%	49.44%	49.90%
Recall	**65.81%**	59.62%	57.16%	49.16%	**72.59%**	47.49%	48.79%	43.99%
F1	**64.74%**	60.00%	58.16%	51.68%	**61.03%**	50.85%	49.11%	46.76%

6. Conclusion

The study of applying reinforcement learning for customer churn prediction using supervised learning dataset is presented in this paper. A public telecommunication dataset from Kaggle is selected. The three classifiers: XGBoost, Random Forest, and kNN are selected and implemented with scikit learn libraries. In literature, XGBoost is currently claimed as the most potent classifier. Random forest is selected as it is ensemble technique and the algorithm embraces the feature of attribute subset selection. The kNN classifier is considered Lazy learner, and it suffers curse of dimensionality. To avoid the issues of sparse data and irrelevant attributes, the experiments were carried out using around twenty attributes obtained from the Random forest. The Lazy learner kNN provides local approximation computed from the k-Nearest Neighbors. Therefore, the model would be less sensitive to the effect of pattern changes, observing that less deviation of the measured values is reported compared to the other models. The preliminary results show that DQN yields the best performance in accuracy, precision, recall, and F1. Besides, DQN could adapt with varied data, and the number of training episodes significantly improves the prediction performance.

7. References

[1] Wertz J 2018 Don't spend 5 times more attracting new customers, nurture the existing ones. https://www.forbes.com/sites/jiawertz/2018/09/12/dont-spend-5-times-more-attracting-new-customers-nurture-the-existing-ones/#519f48e75a8e

[2] Rai S, Khandelwal N and Boghey R 2019 Analysis of customer churn prediction in telecom sector using cart algorithm. In: *Advances in Intelligent Systems and Computing – volume 1045*, pp 457-66

[3] Bhatnagar A and Srivastava S 2019 A Robust Model for Churn Prediction using Supervised Machine Learning. In: *IEEE 9th International Conference on Advanced Computing (IACC)*, pp 45-9

[4] Keramati A, Jafari-Marandi R, Aliannejad M, Ahmadian I and Mozaffari M 2014 Improved churn prediction in telecommunication industry using data mining techniques. In: *Applied Soft Computing - volume 24*, pp 994-1012

[5] Lopez-Martin M, Carro B and Sanchez-Esguevillas A 2020 Application of deep reinforcement learning to intrusion detection for supervised problems. In: *Expert Systems with Applications – volume 141*

[6] Sutton R.S and Barto A.G 2018 *Reinforcement Learning: An Introduction. second edition*. (Cambridge, MA: The MIT Press)