A MapReduce-based Artificial Neural Network Churn Prediction for Music Streaming Service

Min Chen

Department of Computer Science, State University of New York at New Paltz, New Paltz, NY, 12561 US

Summary

Churn prediction is a critical long-term problem for many business like music, games, magazines etc. The churn probability can be used to study many aspects of a business including proactive customer marketing, sales prediction, and churn-sensitive pricing models. It is quite challenging to design machine learning model to predict the customer churn accurately due to the large volume of the time-series data and the temporal issues of the data. In this paper, a parallel artificial neural network is proposed to create a highly-accurate customer churn model on a large customer dataset. The proposed model has achieved significant improvement in the accuracy of churn prediction. The scalability and effectiveness of the proposed algorithm is also studied.

Key words:
Music Streaming, Churn Prediction, MapReduce, Artificial Neural Network

1. Introduction

Predicting the probability that a customer will churn has an important role in the design and implementation of customer retention strategies, especially in saturated industries like the financial or telecommunications sectors. One of the reasons is that in such industries the potential customer base of the (relevant) market is close to be fully allocated between the different competitors. Therefore, the value associated with customer retention tends to be larger than the value obtained from acquiring new customers, which in turn fosters the development of churn management strategies [17].

Churn prediction [1] is a critical long-term problem for many business like music, games, magazines etc. The term ”churn” is defined as not renewing a subscription within a certain period of its expiry. The churn probability can be used to study many aspects of a business including proactive customer marketing, sales prediction, and churn-sensitive pricing models. Hence, a slight improvement in accuracy can lead to dramatic improvement in profit. Since the rise of digital channels for media distribution, music streaming services become popular to customers who want to reduce their expenditures. KKBox [2] is one of the leading music streaming services that offers over 40 million songs to its subscribers by advertising and paid subscription. It is only available to people in select countries in Southeast Asia and Japan. The challenge of the KKBOX is to predict based on users' listening habits whether users will renew their subscription before it expires, or allow it to expire.

Although the existed churn models in the business domain have been used for decades, the modern machine learning models have advanced in recent years. One of the reason is the sheer volume and increasing complexity of the subscription data being collected and created in the field of music streaming services. The conventional churn models are inadequate to handle very large data in a timely manner.

Another reason is the development of big data solutions which enables the processing of a massive volume of data in parallel with many low-end computing nodes. MapReduce [3] and its variants have gained significant momentum from industry and academic because of its simplicity, scalability and fault tolerance. This programming paradigm is a scalable and fault-tolerant data processing tool that was developed to provide significant improvements in large-scale data-intensive applications in clusters. Spark is another popular computing framework for large scale data analytics. It is designed mainly for iterative jobs. Unlike MapReduce, Spark [4] implements in-memory data structures called Resilient Distributed Datasets (RDDs)to cache intermediate data across a set of nodes. Due to the feature of RDDs, algorithms implemented on Spark can be easily iterate over RDD data.

In this work, a parallel version artificial neural network is conducted in a distributed framework provided by the Hadoop/Map-Reduce and Spark infrastructure. This work aims to enhance the computational power and the scalability of parallel artificial neural network in churn prediction. Moreover, it also aims to improve the accuracy of the churn probability to maximize the profit. This is an extension of the work published in [5].

Manuscript received December 5, 2021
Manuscript revised December 20, 2021
https://doi.org/10.22937/IJCSNS.2022.22.1.9

The rest of the paper is organized as follows: literature review is introduced in Section 2. The proposed algorithm is demonstrated in Section 3. The experimental results and analysis are described in Section 4, and the paper is concluded in Section 5.

2. Related Work

The literature on empirical churn modeling in static settings using cross-sections of data is well studied. An extensive benchmark of classification techniques using several real-life data sets is provided in [14] and alternative techniques to analyze churn modeling includes survival analysis are introduced in [7].

The reason why the churn data cannot be directly used is that a majority of the classification methods requires one observation per customer but when there are time-varying features one can usually follow the behavior of the same customer over time and the estimation classification methods cannot directly exploit this type of information [8]. A novel framework called hierarchical multiple kernel support vector machine that without transformation of time-varying features improves the performance of customer churn prediction compared to SVM and other classification algorithms in terms of AUC and Lift using data sets from the Telecom is introduced in [9].

The rising popularity of deep neural networks methods for sequential data has fostered an increase in their applications to churn modeling. A LSTM is taking into account the time-varying features in [10] and the performances are as well as aggregating this information using their average and a random forest algorithm. A combination of CNN and LSTM is proposed to outperforms them individually as well as other algorithms that do not use sequential data in terms of AUC, precision-recall, F1-score, and Mathews correlation coefficient in [11]. Finally, the study in [12] combines different network architectures to leverage the sequential data and show that this combination outperforms CNN, LSTM and classifiers that do not use the time-varying information like random forest and extreme gradient boosting.

3. Proposed Approach

3.1 Data Preparation

The dataset has 482 million entries. The first step is to split it up into readable chunks of fewer than one million lines of data for testing purposes. Algorithm 1 is used to split the data into multiple data files. Each data file contains 5 million entries.

Algorithm 1 Algorithm for Data Partition

Input: A large data file
Output: Multiple files of 5 million lines
1: **while** the input file has more lines **do**
2: **for** $i = 0$ to 5 million **do**
3: Read each data line
4: Write/append to output file
5: **end for**
6: **end while**

For each smaller data file, a group by member number algorithm (see Algorithm 2) is used to further delete the duplication of the data entries.

Algorithm 2 Algorithm for Duplication Removal

Input: A large data file
Output: Data file with unique identifiers
while the input file has more lines **do**
2: read each line and split using commas
$identifier = line[0]$
4: **if** identifier = next identifier **then**
add next line to next identifier
6: remove current line
end if
8: **end while**

3.2 Data Cleaning using Mapreduce

A MapReduce based algorithm is designed to further clean the data. MapReduce framework has two steps: map step and reduce step. The map step and reduce step are listed in Algorithm 3 and Algorithm 4, respectively. The map step import the data from dataset to examine the user log and split it up into its lexicon form. A lexicon is a dictionary which contains a key and its vectors. The reducer would then take the data it receives and group all the data by the unique identifier. By the end of the data cleaning process, the data has been grouped together by its member id.

Algorithm 3 Mapper Function

Input: (key, value)
Output: (key', value')
while the input file has more lines **do**
Read each line and split it up into its lexicon form
3: emit(lexicon, value')
end while

Algorithm 4 Reducer Function

Input: (key, value), import from tensorflow
Output: (key', value')
  for for each line in file **do**
    read key and value of each line
    for items in group **do**
      match key to key' from item
      $counter++$
      $value' += lineValue$ output (key', value')
    end for
  end for

3.3 Artificial Neural Network with Spark

The clean and ready data is used to test the proposed artificial neural network [13] algorithm implemented on Spark. In the input layer, the pre-processed input data is given to neural network model, initially. Each neuron in this layer is connected to neurons of first hidden layer through a synapse which has a weight associated with it. Multiplication of input data value and weight with the addition of bias for each neuron is fed to first hidden layer. The calculation for the subsequent layers is:

$$Y = \sum (weight \times input) + bias \qquad (1)$$

An activation function is used to decide whether the neuron should fire or not. For given input value, the activation function gives the output depending on the function. Using the same process, data flows through the network, layer by layer, until it reaches the output layer. In this work, both sigmoid function and Rectified Linear Unit (Relu) function are utilized.

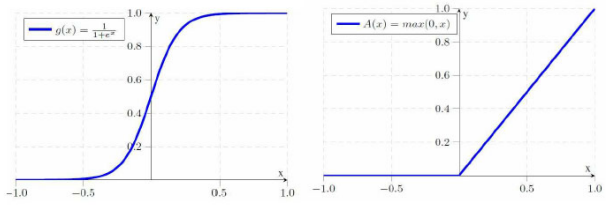


Figure 1: Sigmoid function and the Rectified Linear Unit (Relu) function [6].

The expected output is either 0 or 1. 0 means user is not going to churn and 1 means user is going to churn. The proposed algorithm uses the cross-entropy cost function for the neuron by the calculation:

$$C = \frac{-1}{n} \sum [yln(a) + (1-y)ln(1-a)] \qquad (2)$$

where n is the total number of items of training data, and x and y is the corresponding desired output. Finally, the backpropagation algorithm computes the gradient for each output node and hidden node. These gradients are a measure of how far off, and in what direction (positive or negative) the current computed outputs are relative to the target outputs. The gradients are then used to adjust the values of the neural networks weights and biases so that the computed outputs will be closer to the target outputs. With different optimization algorithms, Adam Optimization algorithm [15] is best for adjusting the weight of the proposed algorithm.

The training model and testing model are conducted on a Spark cluster using MapReduce. The map step trains the neural models with separated data and combine them into one neural network model by taking the average of all the parameters of the neural network models. The detail steps is listed in Algorithm 5. The fully trained neural network model is used to test against the testing model in Algorithm 6.

Algorithm 5 Training Step for the Proposed MapReduced-based Artificial Neural Network

Input: List of (key', value') from User logs
Output: Classifier results from training data
  Define neurons for each hidden layer
  initialize number of classes, the batch size and the number of epochs
  Define first hidden layer with the predefined parameters.
  for i from 2 to n **do**
    update the i^{th} hidden layer using the $(i-1)^{th}$ hidden layer
    update the activation function
  end for
  return the n^{th} layer information

Algorithm 6 Test Step for the Proposed MapReduced-based Artificial Neural Network

Input: Test data
Output: Accuracy
  for Each line of data **do**
    $prediction = neural_network_model(line)$
    $cost = costFunction(prediction, labels)$
    $optimizer = optimizeFunction(learningrate, cost$
    for each epoch **do**
      $epoch_loss = 0;$
      $i = 0;$
      while i ¡ the size of the data set **do**
        $start = i$
        $end = i + batchSize$
      end while
    end for
  end for

4. Experimental Consideration

4.1 Dataset Description

The dataset provided by KKBOX is adopted from the WSDM Cup 2018 Challenge [16]. The dataset consisted of subscriber data from three distinct sources: user activity logs, transactions, and member data. Three years of historical data were included. User log data included a variety of information about subscriber activity by day, transaction data covered all payment and subscription information including renewals and cancellations, and member data contained demographic information about each subscriber such as birth date and gender.

All 3 data sources contained temporal elements, with user activity logs and transactions being a time-series and member data containing initial registration date of the subscriber as well as birthdate. Target variable for our model was the ischurn field, which is a binary label generated via a provided Scala script. The criteria is defined as true if no renewal activity took place within the 30 days of a members subscription expiration date (provided in transactions file). The detail information of the datasets is described in Table 1.

Table I: Information of the datasets.

Dataset	List of Features
Train.csv	msno
	is_churn
Transcations.csv	payment_method_id
	payment_plan_days
	plan_list_price
	actual_amount_paid
	is_auto_renew
	transaction_date
	membership_expire_date
	is_cancel
User_logs.csv	date
	num_25
	num_50
	num_75
	num_985
	num_100
	num_unq
	total_secs
Members.csv	city
	bd
	gender
	registered_via
	registration_init_time

4.2 Experimental Environments

The experiments are conducted on m5.xlarge EC2 (Spark 2.4.4 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1) instances provided by Amazon Web Service. A scalability analysis of the proposed algorithm is conducted with the data nodes range from 1 to 8.

4.3 Experimental Results

The experiments are conducted on the proposed MapReduced-based Artificial Neural Network (MR-ANN) without optimization and (MR-ANN) with Adam Optimization. The model performance was evaluated using a standard log loss calculation on the target variable (churn probability):

$$Log(Loss) = \frac{-1}{N}\sum_{i=1}^{N}(y_i log(p_i) + (1-y_i) log(1-p_i)) \quad (3)$$

The loss function and training accuracy are listed in Figure 2 and Figure 3. The results indicate that the proposed algorithm with Adam optimization have less information loss and better accuracy. Moreover, the results show that the proposed algorithm with Adam optimization converges earlier than the ANN without optimization.

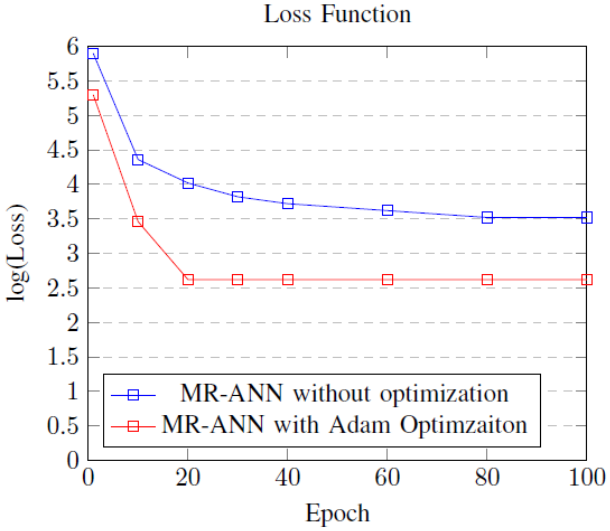


Figure 2: Log(Loss) with different epoches.

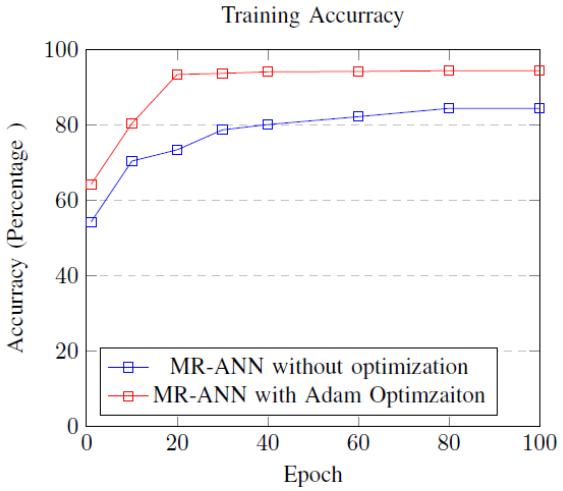


Figure 3: Training accuracy with different epoches.

The execution time of ANN without optimization and ANN with Adam optimization in a single node is demonstrated in Figure 4. The execution time of ANN with Adam optimization does not require significant extra time by comparing to the ANN without optimization. However, due to improved accuracy of the proposed ANN with Adam optimization. The sacrifice of execution time is worthy. The execution time of the proposed ANN with Adam optimization is tested on different data nodes. As the number of nodes increase, the extra experimental become trivial. The Loss seemed to stabilize around 420 out of 5,000,000. The overall accuracy is 94.35%.

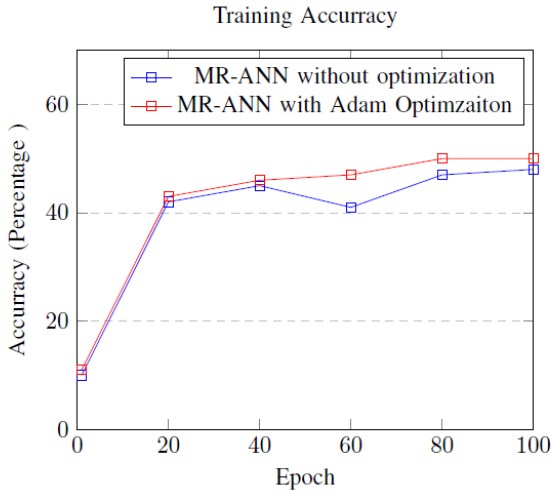


Figure 4: Comparison of execution time with different epoches in a single node.

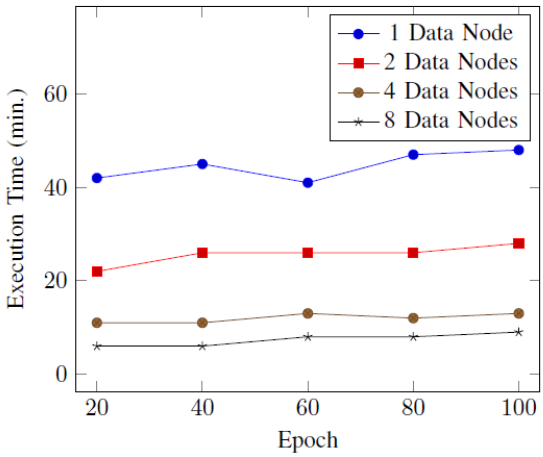


Figure 5: Execution time of different epoches with different data nodes.

5. Conclusion

In summary, a MapReduce-based artificial neural network is proposed to predict customer churn. The proposed algorithm takes advantage of the parallelism of ANN, its rapid convergence and the virtue of the MapReduce paradigm and Spark computing framework. Final accuracy was further boosted by adopting the Adam optimization.

In addition, the overall accuracy achieved on the test dataset is over 94%, which is very significantly out-performed known benchmarks, demonstrating that forecasting KKBOX subscriber churn with a significant level of accuracy is achievable using the methods described in this paper.

The overall accuracy outscored the existed known benchmarks. Future work includes further parameter optimization of the proposed algorithm and further exploration of additional feature engineering not yet tested.

References

[1] Tsai, C.F. and Lu, Y.H., 2009. Customer churn prediction by hybrid neural networks. Expert Systems with Applications, 36(10), pp.12547-12553.

[2] Chen, C.M., Tsai, M.F., Lin, Y.C. and Yang, Y.H., 2016, September. Query-based music recommendations via preference embedding. In Proceedings of the 10th ACM Conference on Recommender Systems (pp. 79-82). ACM.

[3] Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), pp.107-113.

[4] Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S. and Stoica, I., 2010. Spark: Cluster computing with working sets. HotCloud, 10(10-10), p.95.

[5] Chen, M. (2019, October). Music streaming service prediction with MapReduce-based artificial neural network. In 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0924-0928). IEEE.

[6] Arora, R., Basu, A., Mianjy, P. and Mukherjee, A., 2016. Understanding deep neural networks with rectified linear units.arXiv preprint arXiv:1611.01491.

[7] Van den Poel, D. and B. Larivière (2004): "Customer attrition analysis for financial services using proportional hazard models," European Journal of Operational Research, 157, 196 – 217, smooth and Nonsmooth Optimization.

[8] Wei, C.-P. and I.-T. Chiu (2002): "Turning telecommunications call details to churn prediction: a data mining approach," Expert Systems with Applications, 23, 103 – 112.

[9] Chen, Z.-Y., Z.-P. Fan, and M. Sun (2012): "A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data," European Journal of Operational Research, 223, 461 – 472.

[10] Martins, H. (2017): "Predicting user churn on streaming services using recurrent neural networks."

[11] Tan, F., Z. Wei, J. He, X. Wu, B. Peng, H. Liu, and Z. Yan (2018): "A Blended Deep Learning Approach for Predicting User Intended Actions," 2018 IEEE International Conference on Data Mining (ICDM), 487–496.

[12] Zhou, J., J.-f. Yan, L. Yang, M. Wang, and P. Xia (2019): "Customer Churn Prediction Model Based on LSTM and CNN in Music Streaming," DEStech Transactions on Engineering and Technology Research.

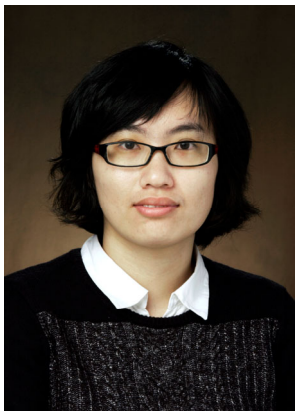
[13] Zurada, J.M., 1992. Introduction to artificial neural systems (Vol. 8). St. Paul: West publishing company.

[14] Verbeke, W., K. Dejaeger, D. Martens, J. Hur, and B. Baesens (2012): "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach," European Journal of Operational Research, 218, 211 – 229.

[15] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[16] WSDM - KKBox's Music Recommendation Challenge https: //www.kaggle.com/c/kkbox-music-recommendation-challenge.

[17] Hadden, J., A. Tiwari, R. Roy, and D. Ruta (2007): "Computer assisted customer churn management: State-of-the-art and future trends," Computers and Operations Research, 34, 2902 – 2917.



Min Chen is now an Assistant Professor at SUNY at New Paltz. She received her bachelor's degree in mathematics and physics from College of St. Benedict in 2009, and earned her master's degree in computer science and doctoral degree in software engineering at North Dakota State University in 2011 and 2015, respectively. Her research interests lie in the area of data science including computer vision, parallel computing, and distributed system. In particular, computational intelligence methods and algorithms are applied to optimization problems in areas such as data mining (including big data) and image processing.