J. Inf. Technol. Appl. Manag. 25(2): 73~90, June 2018
https://doi.org/10.21219/jitam.2018.25.2.073

ISSN 1598-6284 (Print)
ISSN 2508-1209 (Online)

Deep Neural Network Models to Recommend Product Repurchase at the Right Time : A Case Study for Grocery Stores

Hee Seok Song*

Abstract

Despite of increasing studies for product recommendation, the recommendation of product repurchase timing has not yet been studied actively. This study aims to propose deep neural network models usingsimple purchase history data to predict the repurchase timing of each customer and compare performances of the models from the perspective of prediction quality, including expected ROI of promotion, variability of precision and recall, and diversity of target selection for promotion. As an experiment result, a recurrent neural network (RNN) model showed higher promotion ROI and the smaller variability compared to MLP and other models. The proposed model can be used to develop a CRM system that can offer SMS or app-based promotionsto the customer at the right time. This model can also be used to increase sales for product repurchase businesses by balancing the level of ordersas well as inducing repurchases by customers.

Keywords : Purchase Timing, Recommender, Multilayer Perceptron, Recurrent Neural Network, Retail Business, Product Repurchase, Promotion System

Received : 2018. 05. 27.　　Revised : 2018. 06. 18.　　Final Acceptance : 2018. 06. 18.

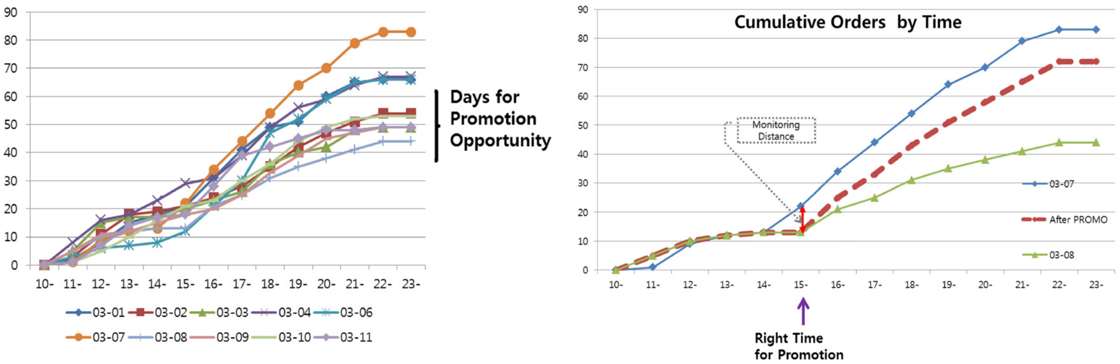
※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2015R1D1A1A09057672).

* Department of Global IT Business in Hannam University, 133, Ojungdong, Daedukgu, Daejeon City 34430, Korea, Tel : +82-42-629-3344, e-mail : hssong@hnu.kr

1. Introduction

With advances in machine learning technologies, recommendation systems are evolving to be smarter and higher quality systems. Existing studies for recommenders have focused on recommending an appropriate item based on customer preference. However, despite its importance, the recommendation of product repurchase timing has not yet been studied actively. In particular, in the case of repetitive purchasing of items such as food or grocery items, there exists many opportunities to increase the sales of a company if the appropriate promotion can be provided at the right time. Sato et al. [2015] and Zhao et al. [2014] also pointed out that research on timing in recommending items is still in its early stages. If a company captures the exact reorder time based on the buying behavior of each customer and sends a promotion offer for a product at the correct time when there is the lack of an order, the company can supplement additional orders, as well as prevent buying from a competitor's company. <Figure 1> shows the cumulative order quantities by hour for a pizza store. There are insufficient order quantities for 7 days out of 10 days, from Mar. 1 to Mar 11,2017, as seen in <Figure (a)>. This provides an opportunity to increase sales if a company sends a promotion offer to target customers who are predicted to purchase soon. The promotion offer can be triggered when the order quantity drops below a predefined order level by monitoring over time, as shown in <Figure (b)>.

The purpose of this study is to propose models for predicting the repurchase time of customers by using only purchase history information consisting of the purchase date of the customers. The prediction model for repurchase timing based on purchase history is advantageous because it can be applied to every product repurchase domain where there exists a history of purchase dates for each customer. The proposed model can be used to build a customer relationship management (CRM) system, including a push-based coupon promotion using mobile SMS or smartphone apps. On the other hand, interest in deep learning has been rapidly increasing due to the success of



(a) Days needed for promotion due to lack of orders (b) Cumulative order quantities by hour

<Figure 1> Orders by Time : The Case of a Pizza Store

AlphaGo, an artificial intelligence Go program made by Google DeepMind. Deep learning is a deeper design of existing neural networks, and it has shown good performance in autonomous navigation, character recognition, natural language processing, etc. However, there is still a lack of research applying deep learning to corporate management. In this study, we apply the recurrent neural network (RNN), one of the most popular deep learning models, to the field of customer relationship management and investigate the usefulness of the model. Specifically, we build a learning model for a grocery domain with the input of weekly purchase patterns, which are obtained by subdividing the purchase history of a week into early-week, mid-week, and weekend, and predict repurchase timing for individual customers based on learned purchase patterns. A multilayer perceptron (MLP) and a RNN are used for predicting the repurchase timing and comparison results of prediction quality including expected ROI of promotion, variability of precision and recall, and diversity of target selection for promotion is presented.

The composition of this study is as follows. Section 2 examines existing methods for predicting the purchase timing and background of recurrent neural networks. In section 3, the structure and input data architecture of MLP and RNN models to predict repurchase timing are proposed. Section 4 summarizes the data set used in the experiment, the performance metrics, and the experimental results. Finally, section 5 introduces a strategy to apply the proposed model to customer relationship management and presents future research directions.

2. Existing Research

2.1 Research to Predict Purchase Timing

In previous research, the way to predict when to buy has been divided into two types depending on what kinds of information are used for prediction : explanatory variable and time series data. Among the approaches to predict when to buy based on an explanatory variable, there are statistical methods and context-aware recommender system using context information. The approach of predicting when to buy using time series data includes the time series model, stochastic probability model, and time interval pattern mining. Agrawal et al. [2011], Oh et al. [2014], and Gould and Dong [2000] tried to predict purchase timing using explanatory variables such as customer characteristics and buying behavior. Agrawal et al. [2011] presented a model to determine the optimal purchase timing for durable goods. In the case of furniture and high-priced electronic products, the price is high at first, but the price tends to decline over time, while the customer has higher satisfaction when purchasing new products, so there exist trade-offs between the benefits of price and satisfaction. They proposed a model that determines the optimal purchase timing by constructing the utility function of customers with the price and the benefit of the customers' preemption as variables. Oh et al. [2014] presented a model for recommending a TV program and developed a recommender system. They found that the start time and the duration of a TV program affected the user's preference for the TV program and built a probability model of watching it to the end using the start time and the duration. The recommendation is made only when there is a TV program with a higher preference than the TV program that the user is currently watching. These two studies are aimed at recommendation timing, but the characteristics of items and the purpose of recommendation are different from this study. One statistical approach to predict repurchase timing by using explanatory variables is a method based on survival analysis. Wang and Zhang [2013] proposed a recommendation system by applying a proportional hazards model in survival analysis. The prediction method of the repurchase time based on survival analysis assumes the cumulative distribution of the purchase probability over time using statistical distribution and predicts the parameters used in the distribution so as to minimize the error of the difference with actual purchase timing data. However, survival analysis is suitable for predicting the repurchase period of durable goods, but is not suitable for predicting the repurchase timing of short purchase cycle products such as food or grocery items.

Meanwhile, the most active research areas for predicting a customer's repurchase timing using time series data are periodic pattern mining, sequential pattern mining, and cyclic pattern mining. These are data mining techniques that detect periodicity, sequential patterns, or cyclic patterns from event data collected in time series. Chiang et al. [2005], Chen and Huang [2005], and Hu et al. [2009] suggested methods to detect time intervals between sequential events. These studies are aimed at discovering repurchase cycles for consumer products by extending sequential pattern mining. Methods of predicting the next purchase time by obtaining the sequential pattern interval have a disadvantage in that it is hard to generalize to a product with an irregular periodicity. The probabilistic model can also be used to predict repurchase timing based on time series data. Rendle et al. [2010] and Wang et al. [2011] proposed a model that predicts the next shopping items based on time series data by combining the latent factor model and the Markov chain model. With the emergence of ubiquitous computing, research on context-aware recommenders has been actively conducted, and these studies also recommend items with timing information [Deng et al., 2014; Yuan et al., 2013; Oku et al., 2009]. A just-in-time recommender provides a function for recommending resources tailored to the preferences and needs of users. It aims at recommending relevant items that match users' personal interests at the right time without waiting for users to initiate any interaction [Akermi et al., 2016]. The context-aware recommender also tries to recommend items at the right time in a specific domain, but it differs from this study in order to develop a general-purpose prediction model for repurchase timing.

With the remarkable success of deep learning, there isan increasing number of studies trying to find deep neural network solutions in the area of recommender systems [Hidasi et al., 2016; Tan et al., 2016]. The insight to apply a deep neural network to recommend items is based on the fact that there are intrinsic patterns in the sequence of customers' buying actions. However, none of the above deep neural network solutions in recommender systems considers the time interval between customers' buying actions, while detecting time interval patterns is crucial for our study to recommend at the right time. Zhu et al. [2017] also suggest that it is important to exploit time information when modeling users'behaviors, so as to improve the recommendation performance.

2.2 Recurrent Neural Network

Recurrent neural network (RNN) is a neural network that mainly refers to in-depth neural networks and improves the expression ability of data by using more intermediate layers than conventional neural networks. Recently, deep learning networks have been rapidly applied in the domain of image recognition and natural language processing with the introduction of a new learning method for in-depth neural networks called pre-learning by Hinton et al. [2006]. RNN is the most popular deep learning model along with convolutional neural network (CNN) and is known to be suitable for learning and reasoning about time series data [Liu et al., 2014; Mulder et al., 2015]. RNN is designed to process the input information based on an understanding of previous input information by combining past and present input information together. Therefore, RNN is mainly used in domains where dependency among input variables is expected to exist, unlike conventional neural networks which mainly use independent input variables as learning data. RNN consists of nodes of input layer (x_{t-1}, x_t, x_{t+1}), hidden layer (s_{t-1}, s_t, s_{t+1}), and output layer (o_{t-1}, o_t, o_{t+1}). RNN also includes weight U that connects the input layer and the hidden layer, weight V that connects the hidden layer and the output layer, and weight W that connects each node of the hidden layer. RNN differs from multilayer perceptron neural networks in that each node of the hidden layer acts as a network memory because the nodes of the hidden layer are sequentially connected so as to easily memorize the previous information. The structure of the RNN can be constructed in various ways. A many-to-many RNN structure is mainly used to produce an output value for every time step in the field of natural language processing such as automatic translation and speech recognition, while a many-to-one RNN structure is used in sentiment analysis that produces a final positive or negative target value when a sentence is entered.

Although the RNN processes the current input information in association with the previous input information, it has a limitation in learning long-range dependencies. According to Bengio et al. [1994], when learning a long sequence of inputs using a back propagation through time (BPTT) learning algorithm, a vanishing gradient problem or an exploding gradient problem may occur. This is a phenomenon where the final gradient value exponentially disappears or increases due to the repetitive multiplication of a small or large gradient along a long sequence in the learning process. To overcome the vanishing gradient problem, it has been suggested to initialize the weights to suitable values and apply a regularization method. The use of the ReLU function instead of the hyperbolic tangent or sigmoid function as the activation function is also proposed as a method to solve the vanishing gradient problem. In addition, there are several RNN models such as long short-term memory (LSTM) or gated recurrent unit (GRU) as fundamental solutions to the vanishing gradient problem. LSTM was first proposed in 1997 [Hochreiter, 1997] and is currently one of the most widely used models in natural language processing, and GRU is a simplified version of LSTM published in 2014 [Chung et al., 2014]. Both models are introduced to overcome the vanishing gradient problem and are known to process long input sequences effectively.

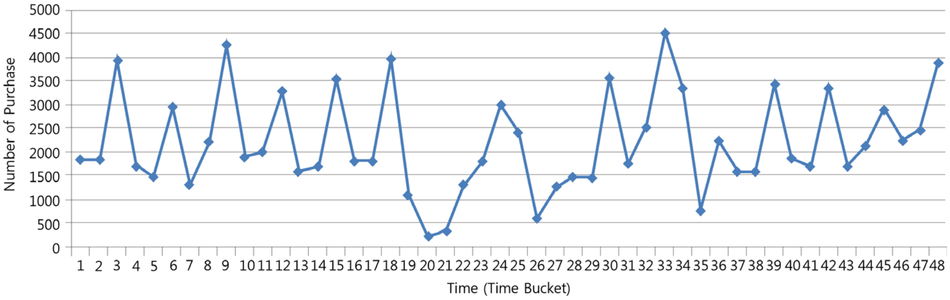
3. Proposed Deep Neural Network Models

The multilayer perceptron (MLP) and recurrent neural network (RNN) are adopted in order to predict the purchase timing of product repurchases by assuming that there are intrinsic periodicity patterns in the customers' buying action. Since the MLP model does not have a mechanism to memorize previous information, it is likely to predict the next purchase timing by learning only the purchase frequency or interval pattern of customers. However, in the case of the RNN model, since it is possible to learn the temporal change in addition to the purchase frequency and the interval pattern for each customer, it may be possible to learn the pattern in which the purchase interval becomes faster or slower.

3.1 Proposed Architecture of Neural Networks

The first decision to construct a neural network model is to design the structure of input data for the model. This study redefines the problem of predicting the repurchase timing of each customer as the problem of predicting who is expected to purchase in the next period depending on the previous purchase history. Since the objective of prediction is to classify "purchase" or "not purchase" in the next period for each customer, binary data is enough to use for training data. Another key decision in relation to the structure design of the input data is to determine the length of the period, which is called a time bucket in this study. Since the suitable length of a period depends on the data characteristics of a specific domain, it is required to examine trends and periodicity in advance. In the Tafeng dataset, which is the purchase history of a grocery store in Taiwan and was adopted in this study to a build model, there is a strong tendency of weekly periodicity, such as purchasing mainly on weekends. <Figure 2> shows the sales trends of the store during four months by dividing it into the time buckets of early-weekday, mid-weekday, and weekend. It shows a strong weekly periodicity pattern where sales volume increases in every weekend bucket. The way the time bucket is divided into early-weekday, mid-weekday, and weekend can be a useful way of providing promotions to prospective buyers. For example, if a store does not have enough orders during a certain time bucket, it is possible to expand the sales volume by predicting target customers who are likely to buy during the time bucket and sending an SMS promotion that offers gifts or discounts.

However, since the RNN model has the capability to learn the temporal change, it may be possible to learn the pattern in which the purchase interval becomes faster or slower over time. Therefore, we define the term "weekly purchase



〈Figure 2〉 Sales Trends by Time Buckets (Tafeng Dataset)

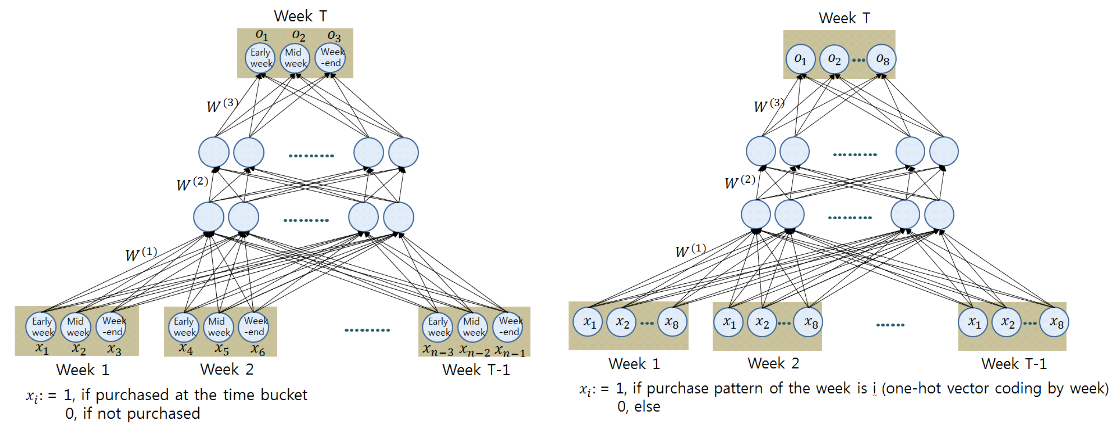
〈Table 1〉 Definition of Weekly Purchase Patterns

Weekly purchase pattern code	Definition of weekly purchase pattern
1	0-0-0 no purchase
2	0-0-1 purchase on weekend
3	0-1-0 purchase during mid-weekday
4	0-1-1 purchase during mid-weekday and on weekend
5	1-0-0 purchase during early-weekday
6	1-0-1 purchase during early-weekday and on weekend
7	1-1-0 purchase during early-weekday and mid-weekday
8	1-1-1 purchase during entire time bucket

pattern" in order to capture the trends of faster or slower purchasing intervals, as well as cyclic trends of heavy purchasing on weekends for each customer at the same time. <Table 1> shows definitions of eight weekly purchase patterns.

The second decision to construct a neural network model is to design the architecture of the model. The architecture of the proposed MLPis shown in <Figure 3>. The MLP to be used for prediction is a four-layered, fully connected neural network consisting of an input, an output, and two hidden layers. The ReLU activation function for the first two layers and the sigmoid activation function for the output layer were adopted for better performance. Figure 3shows two MLP architectures : (a) for the case of not using weekly purchase patterns as input data and (b) the case of using weekly purchase patterns. Both architectures are designed to predict the purchases of T week using the purchase history up to T-1 weeks.

As shown in <Figure 3 (a)>, in the case of the MLP model without weekly purchase patterns, the output value is shown as the probability value for the early-weekday, mid-weekday, and weekend purchases because the sigmoid activation function is used in the final output layer. Therefore, if the output value is greater than or equal to a predefined threshold, the final output is set to 1 (purchase). For the case of the MLP model with a weekly purchase pattern <Figure 3 (b)>, eights neurons in the output layer represent the probability distribution about eight weekly purchase patterns because one-hot vector coding is



(a) MLP architecture without weekly purchase pattern (b) MLP architecture with weekly purchase pattern

〈Figure 3〉 MLP Architecture to Predict Whether to Purchase at T[th] Week

used to represent the weekly purchase pattern code.

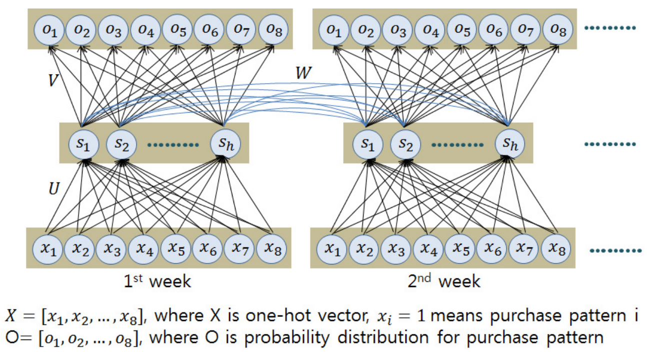
On the other hand, the RNN model has the advantage of reflecting time-dependent characteristics. The final output of the RNN model proposed in this study is expressed as a joint probability of weekly purchase patterns and is calculated as a product of the probability that a purchase pattern in the next period will appear when the purchase patterns of the previous periods are known.

$$P(b_1, \cdots, b_T) = \Pi_{i=1}^{T} P(b_i | b_1, \cdots, b_{i-1}), \qquad (1)$$

where b_i is purchase pattern of i^{th} week

The proposed architecture of the RNN consists of nodes of input layer (x_{t-1}, x_t, x_{t+1}), hidden layer (s_{t-1}, s_t, s_{t+1}), and output layer (o_{t-1}, o_t, o_{t+1}) and includes weight U that connects the input layer and the hidden layer, weight V that connects the hidden layer and the output layer, and weight W that connects each node of the hidden layer <Figure 4>.

The activation function, which is mainly applied to the hidden node, is a hyperbolic tangent (tanh) function that outputs a value between -1 and 1. The activation function applied to the output layer node is a soft max function that converts



〈Figure 4〉 Proposed Architecture of RNN Model the output value into a probability value [Mulder et al., 2015]. In <Figure 3>, the node output of the hidden layer (S_t) and the output layer (O_t) are calculated as follows [Mulder et al., 2015].

$$s_t = \tanh(Ux_t + Ws_{t-1}), \quad \text{where } \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2}$$

$$o_t = \mathrm{softmax}(Vs_t), \quad \text{where } \mathrm{softmax}(x) = \frac{e^x}{\Sigma e^x} \tag{3}$$

Like the MLP model with weekly purchase patterns <Figure 2(b)>, the final value of the output layer is given as the probability distribution for the eight purchase patterns, and the index with the greatest probability is selected as the final purchase pattern of the next week.

3.2 Learning and Implementation of Neural Networks

Learning in the neural network model is doneto find the connection weights U, V, W that minimize the loss function, which is a function of the difference between the target value and the output value. The most common loss function for a neural network is a cross-entropy function. If there are N learning cases, the loss of the predicted value (O_n) of the network to the actual target value (y_n) is calculated as follows.

$$E = -\frac{1}{N}\Sigma_{n\in N}\, y_n \log o_n \tag{4}$$

This study also uses the cross-entropy function as a loss function for the learning of MLP and RNN models. In particular, in an RNN model that produces output values at each time step, the total loss value is calculated as the sum of the losses per time step. The back propagation algorithm (BP), which is a way to efficiently calculate the gradients starting from the output, is used for learning MLP [Nielsen, 2015] and the back propagation through time (BPTT) algorithm was adopted to train the RNN model. Unlike the learning in the BP, the loss function of BPTT is composed of summing up the error values calculated over multiplesteps because the prediction values are calculated separately for each period. The BPTT algorithm is also different from BP in that the gradient at each output depends not only on the calculations of the current time step, but also the previous time steps, because the parameters are shared by all time steps in the network. For example, in order to calculate the gradient at t = 4, it would be required to back propagate three steps and sum up the gradients. The loss function and the gradient are calculated by summing the loss value and the gradient value at each step, respectively, as follows.

$$E = \Sigma_{t=1}^{T} E(t),\ \frac{\partial E}{\partial W} = \Sigma_{t=1}^{T} \frac{\partial E(t)}{\partial W} \tag{5}$$

It is known that RNNs trained with BPTT have difficulties learning long-term dependencies due to what is called the vanishing/exploding gradient problem. The vanishing gradient problem can be avoided by setting an appropriate initial value for the connection weights. It is recommended to set the initial value of the weights at random between $-\frac{1}{\sqrt{n}}$ and $\frac{1}{\sqrt{n}}$

when n is the number of incoming connections from the previous layer [Glorot and Bengio, 2010]. Therefore, initialization of connection weights is performed by applying this method in this study. On the other hand, learning is performed by using stochastic gradient descent (SGD) as a method for finding connection weights in BPTT. SGD is a method of updating the weights by calculating the gradient that minimizes the loss function using randomly sampled data from the training data to reduce computational load for evaluating the sums of gradients. The Theano library [Bastien et al., 2012; Bergstra et al., 2010] was adopted to implement MLP and RNN models in this study. Theano is a Python-based library that is widely used as a commercial service development language for various platforms.

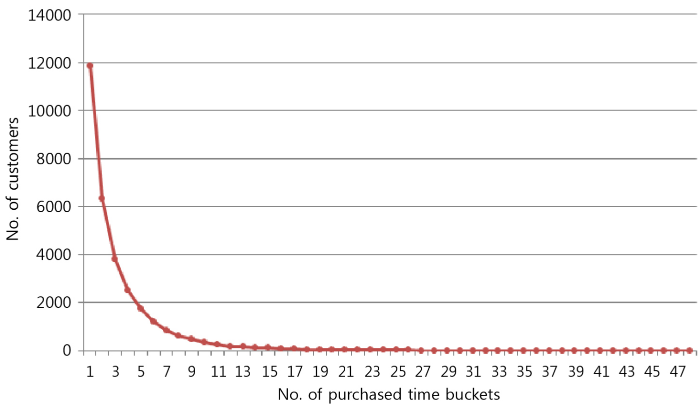
4. Performance Evaluation

4.1 Data Set for Experiment

This study used the Tafeng dataset to evaluate performance of proposed MLP and RNN architectures. The Tafeng data set was collected for research purposes by the Academia Sinica Artificial Intelligence for Investigating Anti-cancer Solutions Lab in Taiwan. This data consists of the purchase history of 817,741 transactions, including 32,266 users with 23,812 items and 2,012 item category information during 4 months (120 days) from November 2000 to February 2001. The data set is divided into 16 weeks, and each week is subdivided into three time buckets again such as early-weekday, mid-weekday, and weekend. Each time bucket has value 1 if one or more purchases were made during this period by each customer, or 0 if there were no purchase transactions during this period for each customer. As a result, the average number of purchased buckets for all customers over four months is 3.358 among 48 time buckets. <Figure 5> shows the distribution of number of customers according to the number of purchased buckets.

4.2 Performance Metric for Evaluation

Precision and recall were adopted to evaluate the prediction quality of MLP and RNN, because



〈Figure 5〉 Distribution of Number of Customers According to the Number of Purchased Buckets the number of purchased buckets and non-purchased buckets are unbalanced in the data set. Precision is defined as the percentage of actual purchasers who are predicted to purchase during the next time bucket by neural network models. Recall is defined as the percentage of predicted purchasers who are actual purchasers in the next time bucket. The number of target customers to be promoted can be another candidate performance metric for practical application, because the number of target customers is related to the number of additional orders. If there are too many target customers for promotion, delivery may be delayed due to order congestion, and if the number of promotional target customers is too small, the promotional effect may hardly appear. Once the number of target customers for promotion has been determined, the expected return on investment (ROI) for the promotion can be calculated in the following manner, since the precision has already been calculated.

$$\begin{aligned}\text{Expected ROI} = &(\mathrm{P} \times \mathrm{EP}) - (\mathrm{TC} \times C_{sms}) \\ &- (EP \times C_{gift}) - (TC - EP) \\ &\times C_{spam}\end{aligned} \tag{6}$$

where, TC : The number of target customers for promotion

EP : Expected purchasers (= TC×Precision)

C_{sms} : Cost per SMS

C_{spam} : Cost per spam SMS

C_{gift} : Cost of promotion gift per purchaser

P : Profit for unit sale

In addition to the expected ROI, it is necessary to review the variability of precision and recall in order to apply the proposed model to practical industry. If the performance evaluation results of the model show a lot of differences in each turn, the model may not be reliable. The variability is calculated using the average deviation of the precision and the recall for each prediction in this experiment. Another performance metric is the diversity of target customer selection. If the same customer is selected for promotion every week, it will be difficult to apply to actual marketing promotions. Therefore, the metric (D2) of how different the target customers are for each of the early-weekday, the mid-weekday, and the weekend, and the ratio of the customers (D1) who were promoted in the previous week can be used to determine diversity of target customer selection, as follows.

$$\mathrm{D1} = 1 - \frac{\| C_{i-1} \cap C_i \|}{\| C_i \|} \tag{7}$$

$$\mathrm{D2} = 1 - \frac{\| C_i^{ew} \cap C_i^{mw} \cap C_i^{we} \|}{\| C_i \|} \tag{8}$$

C_i : Target customers for promotion during i^{th} week

C_i^{ew} : Target customers for promotion during early weekday of i^{th} week

Model learning and performance evaluation are done in the following way. First, the neural network model is learned by using the purchase history from the 1^{st} week to the $T-2^{th}$ week as the explanatory data, assigning the purchase history of the $T-1^{th}$ week as the target data. When the learning is completed, the purchasing history from the 2^{nd} week to the T-1^{th} week is used as explanatory data to predict whether the customer will purchase during the T^{th} week, and then the performance evaluation is performed comparing it with the actual purchase. In this way, the prediction from T week to T+N week is repeated and the N+1 prediction results are averaged to be used as a final performance measurement.

4.3 Experiment Results

A naïve model was constructed to set the base for comparison in the performance evaluation. In the naïve model, it is predicted to purchase during T-1^{th} week if the number of purchased time buckets from the 1st week to T-2^{th} week is greater than the threshold value. The threshold value is determined as a value that maximizes the F-measure for the predicted result of the T-1^{th} week. Then, if the number of time buckets from the 2^{nd} week to T-1^{th} week is greater than this predetermined threshold value, it is predicted to purchase during T^{th} week, as follows.

$$o_T^k = \begin{cases} Purchase, & if\ \sum_{i=T-p}^{T-1} x_i^k \geq \theta_k^* \\ No\ purchase, & else \end{cases} \tag{9}$$

$$\theta_k^* = \mathrm{argmax}_\theta\, F_{T-1}^k \tag{10}$$

$$F_{T-1}^k = \frac{(\beta^{2+1}) \times Precision_{T-1}^k \times Recall_{T-1}^k}{\beta^{2\times} Precision_{T-1}^k + Recall_{T-1}^k} \tag{11}$$

where, k = early-weekday, mid-weekday or weekend

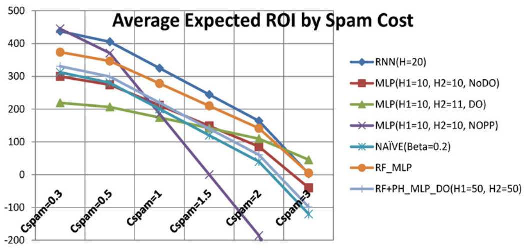
x_i^k = 1, if purchased in the time bucket k of week i (else 0)

o_T^k = 1, if purchased in the time bucket k of week T (else 0)

Precision is an important criterion for determining the spam rate, as well as the response rate, to the promotion in this study. Therefore, it is desirable to obtain the F value by placing a higher weight on precision. We set β to 0.2 in this experiment. The training model was built with the purchase history from the 1^{st} week to 10^{th} week as the explanatory data and the purchase of 11^{th} week as the target data. Then, the purchasing history of the 2^{nd} week to 11^{th} week is input into this model again to predict whether the customer will purchase during the 12^{th} week. In this way, the predictions during 5 weeks from the 12^{th} week to 16^{th} week were performed independently, and then the performance of each model was compared using the 5-week average.

Seven models were prepared for performance evaluation. RNN (H = 20) is an RNN model including a hidden layer with 20 nodes, and MLP (H1 = 10, H2 = 10, NoDO) is a fully connected MLP model with 10 nodes in the first and second hidden layers. MLP (H1 = 10, H2 = 11, DO) is a model in which dropout is applied to 20% of the connections between the input layer and the first hidden layer in the MLP model. Dropout is a regularization technique for neural network models proposed by Srivastava et al. [2014]. MLP (H1 = 10, H2 = 10, NOPP) is an MLP model without using weekly purchase patterns, as shown in <Figure 2 (a)> and NAIVE (Beta = 0.2) is the basic naïve model with beta set to 0.2. In addition, RF_MLP is an MLP model that uses only recency and frequency, which are derived from the purchase history, as input data, and RF+PH_MLP_DO (H1 = 50, H2 = 50) is an MLP model with 50 nodes of the first and second hidden layers and uses purchase history, recency and frequency together as input data. <Figure 5> shows the average number of target customers selected for promotion by each model and the average number of expected purchasers as a result of promotion. The number of expected purchasers was calculated by multiplying the number of target customers for promotion by the precision of each model.

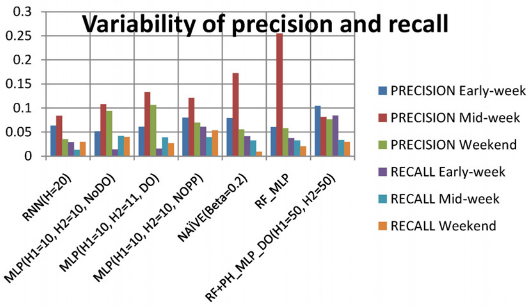
According to <Figure 6>, the average number of target customers for promotion and the average number of expected purchasers are the largest in the MLP (H1 = 10, H2 = 10, NOPP) model, and the smallest in the MLP (H1 = 10, H2 = 11, DO) model. The average expected ROI can be calculated after the average number of target customers for promotion and expected purchasers are calculated. Average expected ROI is estimated under the assumption of $C_{sms} = \$0.2$, $C_{gift} = \$2$, $P = \$4$. Since the cost of spam SMS (C_{spam}) is unknown, the average expected ROI is examined by changing the cost of spam from \$0.3 to \$3. In Figure 7, the average expected ROI of the RNN (H = 20) model is the highest in the interval of $0.3 < C_{spam} < 2.5$. The average expected ROI of the MLP (H1 = 10, H2 = 10, NOPP) model, which does not use weekly purchase patterns, is significantly reduced as the cost of spam increases. Therefore, in this domain, it is better to apply the purchase pattern to predict the customers to be targeted when the cost of spamming is high. The average expected ROI calculated in this experiment was calculated in a very conservative manner. Since the precision used in this experiment was calculated without providing any kind of gift or discount to the target customer for promotion, the actual ROI may be higher if the store provides an appropriate discount or gift for each customer in practical application.



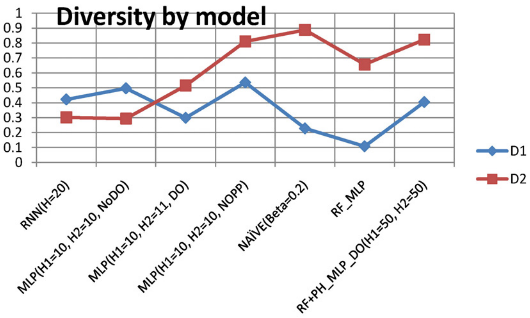
⟨Figure 7⟩ Average Expected ROI by Model and Spam Cost

<Figure 8> compares the performance of each model in terms of variability of precision and recall. As shown in the graph, the variability of the RNN (H = 20) model is the smallest, and is the most stable model. Figure 9 shows diversity by model. The MLP (H1 = 10, H2 = 10, NOPP) model is the most favorable for D1 in terms of

⟨Figure 6⟩ Average Number of Target Customers for Promotion and Expected Purchasers how different customers are selected for promotion each week. For the metric D2 of how different the target customers are for each of the early-weekday, the mid-weekday, and the weekend, the NAIVE (Beta = 0.2) model was the highest.



〈Figure 8〉 Variability of Precision and Recall for 5 Predicted Weeks



〈Figure 9〉 Diversity by Model

On the other hand, if there are restrictions such as delivery capacity or lack of inventory, it is desirable to select the model by considering the number of target customers for promotion rather than expected ROI. In addition, if frequent SMS promotions are thought to be able to give customers fatigue, it is necessary to increase the diversity. The weekly diversity also will be increased when a different model is chosen to predict target customers for promotion every week.

5. Conclusion

Deep neural network models were proposed to recommend product repurchase at the right time based on simple purchase history information, and the usefulness of the RNN model in the grocery domain was investigated through performance evaluation. Performance metrics were also proposed to choose effective models for target selection, and the usefulness of the metrics was checked. The proposed models and metrics can be applied for customized promotions to increase sales in the retail sector. The two profit-making approaches in the retail business are cost reduction and sales growth. However, it is hard to make profit through cost reduction in this competitive business sector because there has already been a lot of pressure to cut costs for a long time. According to the literature of customer relationship management, in terms of sales growth, it is desirable to obtain a large number of loyal customers by attracting repetitive purchase. One way to prevent customer churn to competitors, as well as to promote repetitive buying, is to recommend the right product at the right time. The proposed models and metrics in this study can be applied for the promotion system of retail businesses to expand sales when there is a lack of orders.

Recurrent neural networks (RNN) have been mainly applied to language models until now. However, it is hard to find an RNN model applied to CRM. In the case of the language model, there are performance issues of RNN models such as running time, because the number of vocabulary words is large. On the other hand, in the case of CRM application of the RNN model, performance is not an important issue because the number of dimensions in RNN nodes and weights is relatively small in customer data. In terms of prediction quality, the prediction accuracy of RNN models in CRM tends to be lower than that in natural language processing since it is hard to find distinct patterns in customer data, while there are common patterns in language sentences due to sharing grammars. In addition, the sparseness of purchase history data causes prediction accuracy to deteriorate in the RNN models of CRM. Therefore, it is necessary to use a strategy for aggregating the number of purchases for an appropriate period or designing purchase patterns with additional features to overcome thesparsity issue in purchase history data.

For future research, it is possible to build a personalized promotional system using the deep neural network models proposed in this study. When the response to a promotion offer, which is provided by the customers, is fed back and accumulated, the prediction quality can be improved by using the response to the promotion as an additional input data of the prediction model proposed in this study. Sato et al. [2015] also proposed a customized Bayesian personalized ranking (BPR) recommendation system that combines sensitivity information based on price discounts as well as customer preferences for items in the retail business sector.

References

[1] Agrawal, R., Ieong, S., and Velu, R., "Timing When to Buy", *ACM Conference on Information and Knowledge Management (CIKM)*, 2011.

[2] Akermi, I., Boughanem, M., and Faiz, R., "Just-In-Time Recommendation Approach within a Mobile Context", *IEEE/WIC/ACM International Conference on Web Intelligence*, 2016, pp. 636-639.

[3] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., and Bengio, Y., "Theano : new features and speed improvements", *NIPS 2012 deep learning workshop*, 2012.

[4] Bengio, Y., Simard, P., and Frasconi, P., "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, 1994, pp. 157-166.

[5] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., WardeFarley, D., and Bengio, Y., "Theano : a CPU and GPU math expression compiler", *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[6] Chen, Y. L. and Huang, T. C. K., "Discovering fuzzy time-interval sequential patterns in sequence databases", *IEEE Syst. Trans. Man Cybernet Part B*, Vol. 35, No. 5, 2005, pp. 959-972.

[7] Chiang, D. A., Lee, S. L., Chen, C. C., and Wang, M. H., "Mining interval sequential patterns", *International Journal of Intelligent System*, Vol. 20, No. 3, 2005, pp. 359-373.

[8] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., "Empirical Evaluation of Gated Recurrent

Neural Networks on Sequence Modeling", *NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

[9] Deng, Z., Yan, M., Sang, J., and Xu, C., "Twitter is Faster : Personalized Time-Aware Video Recommendation from Twitter to YouTube", *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 11, No. 2, 2014.

[10] Glorot, X. and Bengio, Y., "Understanding the difficulty of training deep feedforward neural networks", *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, 2010.

[11] Gould, B. W. and Dong, D., "The Decision of When to Buy a Frequently Purchased Good : A Multi-Period Probit Model", *Journal of Agricultural and Resource Economics*, Vol. 25, No. 2, 2000, pp. 636-652.

[12] Hidasi, B., Karat-Zoglou, A., Baltrunas, L., and Tikk, D., "Session-based recommendations with recurrent neural networks", In ICLR, 2016.

[13] Hinton, G. E., Osindero, S., and The, Y., "A fast learning, algorithm for deep belief nets", *Neural Computation*, Vol. 18, 2006, pp. 1527-1554.

[14] Hochreiter, S., "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735-1780.

[15] Hu, Y. H., Huang, T. C., Yang, H. R., and Chen, Y. L., "On mining multi-time-interval sequential patterns", *Data Knowledge Engineering*, Vol. 68, No. 10, 2009, pp. 1112-1127.

[16] Liu, S., Yang, N., Li, M., and Zhou, M., "A Recursive Recurrent Neural Network for Statistical Machine Translation", *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 1491-1500.

[17] Mulder, W. D., Bethard, S., and Moens, M. F., "A survey on the application of recurrent neural networks to statistical language modeling", *Computer Speech & Language*, Vol. 30, No. 1, 2015, pp. 61-98.

[18] Nielsen, A. M., *Neural Networks and Deep Learning*, Online ebook, 2015.

[19] Oh, J., Kim, S., Kim, J., and Yu, H., "When to recommend : A new issue on TV show recommendation", *Information Sciences*, Vol. 280, No. 1, 2014, pp. 261-274.

[20] Oku, K., Nakajima, S., Miyazaki, J., Uemura, S., and Kato, H., "A Recommendation Method Considering Users' Time Series Contexts", *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, 2009, pp. 465-470.

[21] Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L., "Factorizing personalized Markov chains for next-basket recommendation", *WWW Conference*, 2010, pp. 811-820.

[22] Sato, M., Izumo, H., and Sonoda, T., "Discount Sensitive Recommender System for Retail Business", *Proceedings of the 3rd Workshop on Emotions and Personality in Personalized Systems*, 2015, pp. 33-40.

[23] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout : A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, Vol. 15, 2014, pp. 1929-1958.

[24] Tan, Y. K., Xu, X., and Liu, Y., "Improved recurrent neural networks for session-based recommendations", *RecSys ACM*, 2016, pp. 17-22.

[25] Wang, J., Sarwar, B., and Sundaresan, N., "Utilizing related products for post-purchase recommendation in e-commerce", *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 329-332.

[26] Wang, J. and Zhang, Y., "Opportunity Models for E-commerce Recommendation : Right Product, Right Time", *Proceeding of SIGIR'13*, 2013.

[27] Yuan, Q., Cong, G., Ma, Z., Sun, A., and Magnenat-Thalmann, N., "Time-aware Point-of-interest Recommendation", *Proceeding of SIGIR'13*, 2013.

[28] Zhao, G., Lee, M. L., and Wynne, H., "Utilizing Purchase Intervals in Latent Clusters for Product Recommendation", *Proceedings of the 8th Workshop on Social Network Mining and Analysis (SNAKDD'14)*, 2014, pp. 1-9.

[29] Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., and Cai, D., "What to Do Next : Modeling User Behaviors by Time-LSTM", *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 3602-3608.

Author Profile

Hee Seok Song

Hee Seok Song is a professor of management information systems department at Hannam University in Korea. He received bachelor's degree from Korea University and master and Ph.D degree from Korea Advanced Institute of Science and Technology. His research interests include intelligent computing technology, business intelligence, and social network. He published his studies to Knowledge-based systems, Expert systems with applications, Artificial Intelligence Review, and many international and domestic journals.