Technology Reports (Special Articles) Behavior Prediction Recommendations RNN

Special Articles on AI—Expansion of AI Technologies to Diverse Industries and Basic Technologies Supporting AI Applications—

A Recommendation Engine Using Time Series Prediction Models of User Behavior

Service Innovation Department Taku Ito Yuri Sasaki Shigeki Tanaka

The provision of a recommendation function for products and content in B2C services on the web has become a common feature in recent years. Recommendation engines come in various types, but most are based on popularity ranking, which means that they have not been able to truly understand the context of service use by a particular user. To present recommendations that attract the user's interest, a recommendation engine must be able to understand user context and predict content that the user should find interesting. With the aim of raising the probability of purchases, NTT DOCOMO has developed a recommendation engine that interprets behavior based on time series data and applies a deep learning algorithm to make predictions. This engine has been applied to NTT DOCOMO services enabling the provision of highly accurate recommendations.

1. Introduction

It has become common in Electronic-Commerce (EC) services*1 and Business-to-Consumer (B2C) services to provide product recommendations*2. A recommendation function is also being provided in many NTT DOCOMO services as "recommendations for the user." A wide range of recommendation techniques have been proposed from simple techniques based on popularity ranking to those

*1 EC services: Services that enable the buying and selling of goods on the Internet.

using machine learning*3, and in just the last few years, an easy-to-use machine learning library*4 has appeared on the market and recommendation services using cloud services*5 have begun to be provided.

Recommendation functions have been provided for a number of years, but at present, many recommendation systems are based on popularity ranking*6. This approach determines the preferences of a majority of users and can therefore achieve a certain Click-Through Rate (CTR)*7, but it is not without its problems. For example, since the same products are recommended to different users in such a system, those recommendations have no effect on users who have no interest in popular content, and given that the content presented is fixed, many other types of products go without recommendations making it difficult for users to discover new products. Additionally, since the procedure typically followed by a user is to click on desirable products during a sequence of web movements, time series data that can reveal the sort of content the user has just clicked on is more important than interests or preferences in a recommendation service. For example, given a user searching for food on a portal site, it would be desirable to present food against the background of the user's web movements without regard to everyday interests or preferences. If the user has just clicked on "fresh foods," the user can then be presented with fresh foods thereby enabling content desired by the user to be recommended.

Against the above background, NTT DOCOMO has applied a Recurrent Neural Network (RNN) as a time series prediction algorithm to the generation of recommendations so that long-term and short-term shifts in user interests can be determined. This technology mainly makes recommendations in genres that the user has recently showed an interest in but can also make recommendations in genres that the user has regularly been interested in. In this way, recommendations can be presented tailored to personal conditions, which can help improve CTR. In this article, we describe recommendation algorithms that have actually been applied and examine their effectiveness in certain NTT DOCOMO services.

2. Using RNN to Make Recommendations

2.1 RNN Overview

An RNN, which is also called a recursive type of neural network*8, is a neural network model designed to recognize patterns in sequential data such as time series data*9 [1]. The conventional neural network model features a fixed-length input layer and output layer plus middle layers between those two layers (**Figure 1**). In an image recognition task, for example, the actual image can be divided into units of pixels so that pixel values are treated as the input layer while categories as targets of classification are output on the output layer. In this example, the input layer and output layer consists of a number of pixels and number of classification patterns, respectively, with each taking on a single value. This means that each node*10 must be of fixed length.

On the other hand, actual data cannot necessarily be given in a fixed length. For example, in a task like document prediction, a document is given as input, but since a document is not of fixed

*2 **Recommendations:** The process of recommending products and content tailored to the user.

*3 **Machine learning:** Computer algorithms for learning patterns based on input data and executing various types of tasks.

*4 **Library:** A collection of high-versatility programs in a reusable form.

*5 **Cloud services:** The provision via the network of services running on remote servers instead of on the user's computer.

*6 **Ranking:** The listing of products and content in the order of most clicked or purchased over the entire service. It is one of the most basic types of recommendation algorithms.

length as in the case of an image, it cannot be expressed in terms of the above model. An improved model that can handle variable length data is RNN, which handles the points making up variable length data as a time series and stores variables called "states" within the model (in middle layers) (**Figure 2**). The state changes according to input data and that state propagates to the next time series step. In addition, a value corresponding to that state is presented in the output layer. In this way, a model in

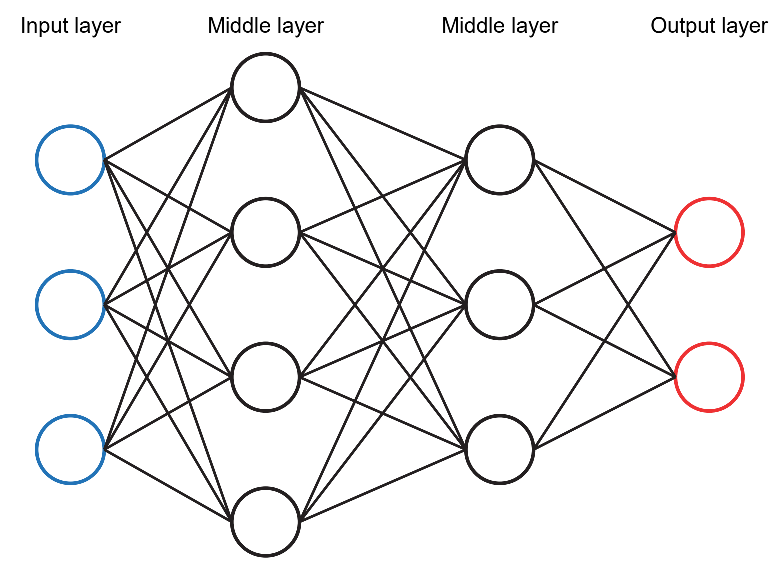

Figure 1　Ordinary neural network

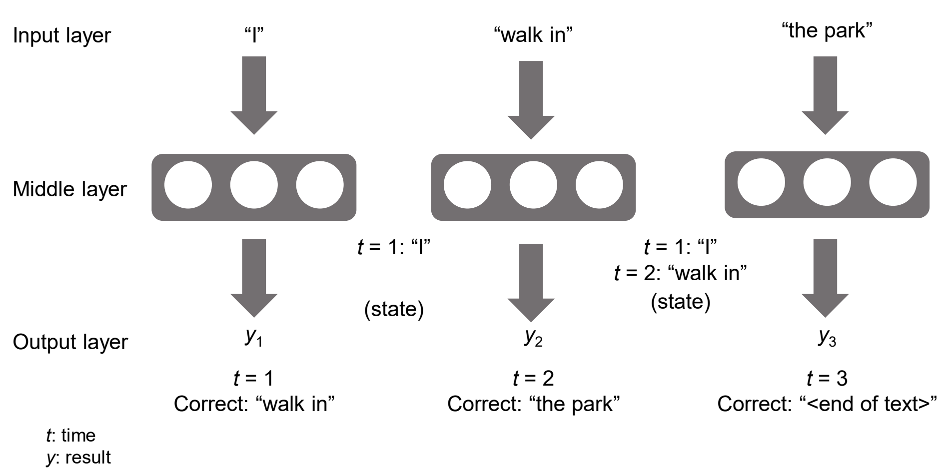

Figure 2　Schematic diagram of RNN

*7　CTR: The ratio of the number of times the user has actually clicked on content to the number of times content has been displayed by a recommendation function.

*8　Neural network: A representation of the neural network inside the human brain by a numerical model consisting of an input layer, middle layers, and output layer.

*9　Time series data: Data containing information on the temporal change in values.

*10　Node: A network point that propagates a value received from input.

which states propagate along a time series makes it possible to apply a neural network to target applications that take variable length data as input.

2.2 Application of RNN to Recommendations

The RNN described above is often used in tasks targeting text such as document recognition*11 and machine translation*12 and tasks targeting speech such as speech recognition. Here, by having a system learn about the data within a network in the manner of "what kind of output value should be given for a given input value," a desired output value with respect to new text or speech can be obtained. Because of this characteristic, RNN can also be used in time series prediction. Training a system in which input and output values are of the same type enables the construction of a model that can predict the next value to come given a specific input [2].

Recent years have seen an increasing number of RNN time series prediction tasks being applied to recommendation systems. In such a system, each input value can be defined as an instance of user behavior such as a product click or video view and an output value can likewise be given as user behavior. In this way, given a user with an interest in a certain product or video, the transition in a user's interest (state), that is, what kind of product or video would that user have an interest in next, can be predicted (**Figure 3**). In addition, since it then becomes possible to make recommendations in line with the user's state at that time, inputting the latest instance of user behavior into the model enables the real-time provision of updated recommendations at all times.

3. Proposed Techniques

3.1 Overview

On applying RNN to the process of making recommendations, we have been making several improvements tailored to the service domain and devising measures to output recommendations applicable to the user's recent interests. In particular, we constructed (1) a model for making recommendations that combines a long-term prediction model and short-term prediction model using hierarchical

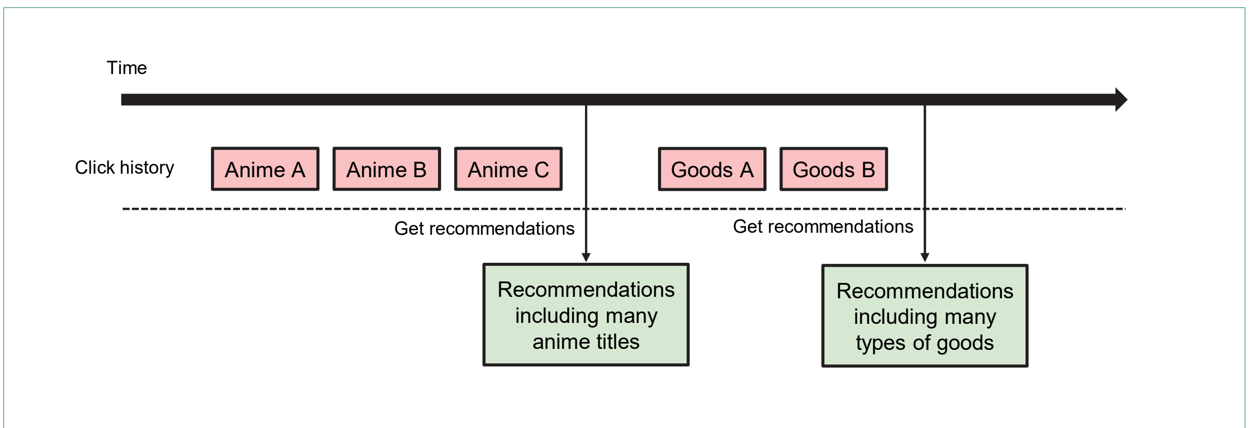

Figure 3 Overview of using RNN for recommendations

*11 **Document recognition:** A task that determines the category that a certain document belongs to.

*12 **Machine translation:** A task that automatically translates a document in a certain language to a document in another language using a computer.

RNN and (2) a model for making recommendations that combines multiple RNNs using different information. These models are explained below.

3.2 Long-term/short-term Prediction Model Using Hierarchical RNN

We can consider that user interests undergo both long-term transitions and short-term transitions as opposed to changes over a fixed period [3]. For example, in a system that recommends smartphone applications (apps) based on the user's app installation history, the case of searching for a Social Networking Service (SNS) app on day X and the case of searching for a video delivery app on day X+1 can be considered. It is therefore desirable that SNS apps be recommended on day X and that video delivery apps be recommended on day X+1. However, a simple RNN cannot consider any differences between day X and day X+1, so the possibility exists that SNS apps will be recommended on day X+1 too on the basis of the user's history on day X. To prevent this from happening, it is necessary to recommend SNS apps during day X based on the user's installation history of that day, and when crossing into a new day, to predict a long-term transition in user interests and make recommendations accordingly. Here, a long-term transition in interests refers, for example, to the searching for a video delivery app on the day following a search for an SNS app.

We propose a hierarchical RNN to achieve recommendations that grasp the user's interests over the long term and short term as described above (**Figure 4**). Specifically, the above problem can be

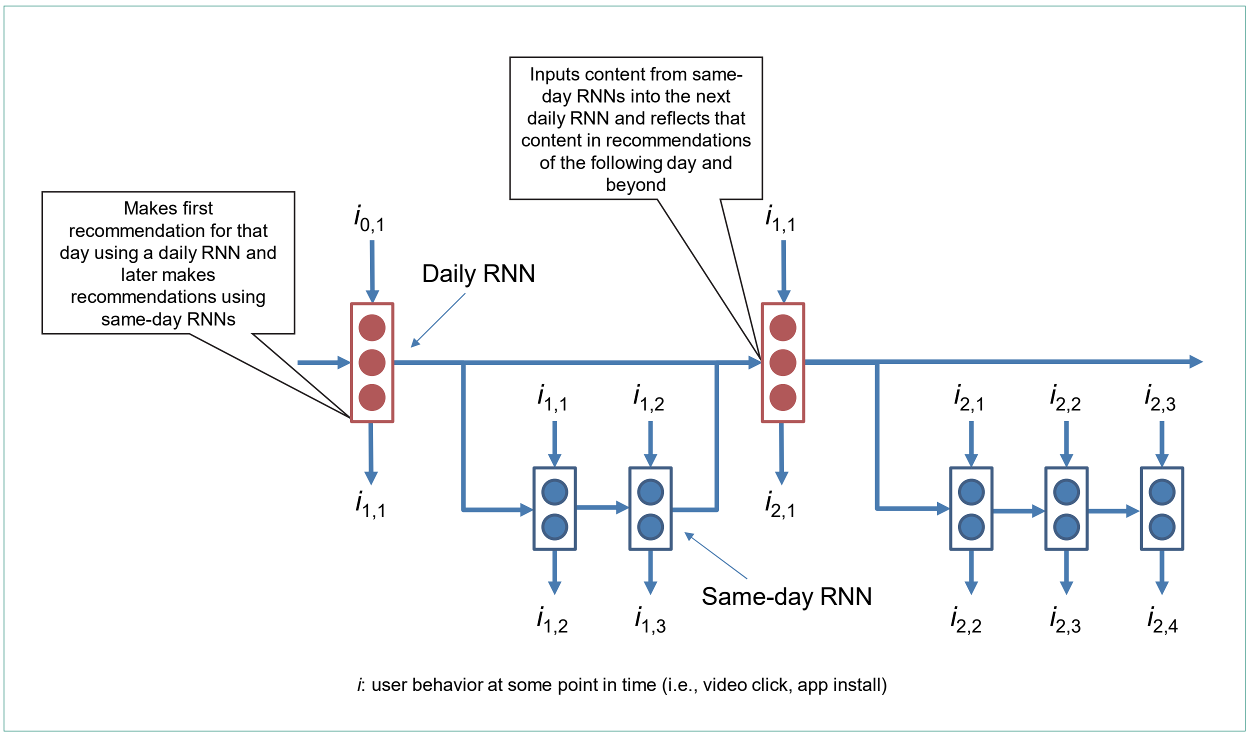

Figure 4 Schematic diagram of hierarchical RNN

solved by dividing the process into a daily RNN, which determines the content for making recommendations from user history up to the previous day, and same-day RNNs, which determine the content for making recommendations from user history of the previous day and current day.

3.3 Recommendation Model that Combines Multiple RNNs Using Different Information

Recommending certain products or videos must be based on the content of such items. On the other hand, the data handled by RNNs consist of numerical values, so it is necessary to handle products, videos, and other goods as numerical data. In general, content IDs are allocated to products, videos, etc., which enables an RNN to predict the time series transition of that ID-based numerical data. With this technique, predictions are made in units of content IDs, but if the same type of content does not exist in the user's past history, no predictions can be made. However, a similar type of time series transition can be considered even in the case of a different content ID as long as similar content exists, so the content (category) of products and video must be taken into account. For example, we can consider that a user who has clicked on a product in the eating-and-drinking category has a high probability of clicking on another product in the same category. It is therefore desirable that predictions be made not only for a time series in units of content IDs but also for a time series in units of categories.

To achieve recommendations in the manner described above, we propose a recommendation model that combines multiple RNNs (**Figure 5**). Combining RNNs that make time series predictions while making data on different layers independent of each other makes it possible to achieve both detailed recommendations in units of content and

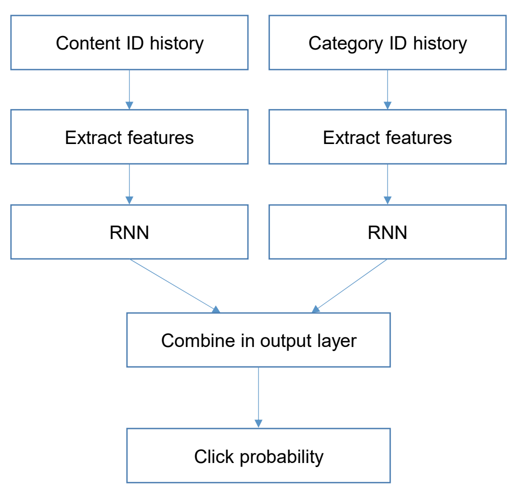

Figure 5 Schematic diagram of multiple RNNs

general recommendations in units of categories thereby improving both accuracy and coverage*13.

4. Application to NTT DOCOMO Services

We tested the effectiveness of applying the above techniques to actual NTT DOCOMO services, namely, "App Recommender" that recommends apps to the user and the NTT DOCOMO-managed portal site called "dmarket" that recommends products of B2C services. We applied the "recommendation model using hierarchical RNN" to App Recommender and the "recommendation model using multiple RNNs" to dmarket. The following gives an overview of each service and presents the results of testing the effectiveness of each technique.

4.1 App Recommender

The service screen is shown in **Figure 6**. This service presents apps thought to be useful for the user based on the user's history of installing apps. With the aim of getting the user to install more of the displayed apps, we tested the effectiveness of the recommendation model using the hierarchical RNN described above with actual users. Specifically, given the proposed technique and an existing recommendation algorithm using user features*14 based on history, we conducted an A/B test*15 of these two methods from September 2020. The proposed technique achieved an install rate*16 2.7 times that of the existing algorithm. The reason for this high install rate can be given as follows. Taking into account time series data, a user who has just installed, for example, Twitter and a camera app in succession could be recommended Instagram.

Figure 6　Service screen of App Recommender service

In other words, the characteristics of apps desired by a user could be inferred from immediately previous user behavior.

4.2 dmarket

The service screen is shown in **Figure 7**. As shown here, this service displays content recommended for the user in a cross-service manner in domains classified by 16 tabs such as "shopping" and "gourmet." The objective here is to display recommendations that will induce the user to click on content of interest spanning multiple services. Additionally, the recommended content can be varied each time it is displayed during web movements to continuously attract the user's interest. We conducted a test using actual users on the effectiveness of recommendations that combine multiple RNNs as described above. Specifically, on conducting an A/B test against this technique and the

---

*13 **Coverage**: The ratio of the total number of content items recommended to all users to the total number of all content items. High coverage indicates that a wide range of content is being recommended.

*14 **Features**: Quantities (numerical values) extracted from data and characterizing that data.

*15 **A/B test**: A test that compares two algorithms to determine which is more effective.

*16 **Install rate**: The ratio of apps that the user has actually installed to all apps presented by a recommendation function.

Figure 7　Service screen of dmarket service

existing recommendation algorithm based on popularity ranking, we achieved a 2.0% improvement in CTR. We consider the reason for this improvement in CTR is that the proposed technique could extract common elements from items of content belonging to different services. For example, many food items can be recommended by popularity ranking on the gourmet tab, but for a user that frequently uses dmagazine*17, many gourmet magazines can be recommended as well. In short, it became possible with the proposed technique to recommend services applicable to that user even for the same gourmet tab.

5. Conclusion

This article described recommendation algorithms as an extension of RNN. It described, in particular, recommendations using hierarchical RNN for determining shifts in user interests in both the long term and short term tailored to service domain characteristics and recommendations that combine multiple RNNs using multiple features of content. Both algorithms were tested for their effectiveness in actual NTT DOCOMO services and were found to lead to improved accuracy compared with existing algorithms. At NTT DOCOMO, we aim to pursue the latest technologies in recommendation algorithms and to promote initiatives that raise the value of service provision.

REFERENCES

[1] J. Li, L. Deng, R. H.-Umbach and Y. Gong: "Robust Automatic Speech Recognition: A Bridge to Practical Applications," Academic Press, Oct. 2015.

[2] D. Wierstra, J. Schmidhuber and F. Gomez: "Evolino: Hybrid Neuroevolution/Optimal Linear Search for Sequence Learning," Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, pp.853–858, Jul. 2005.

[3] B. Hidasi, A. Karatzoglou, L. Baltrunas and D. Tikk: "Session-based recommendations with recurrent neural networks," arXiv preprint arXiv: 1511.06939, Nov. 2015.

*17 dmagazine: A subscription-based magazine delivery service provided by NTT DOCOMO.