Adopted Transfer Learning to Item Purchase Prediction on Web Marketing

Noriko Takaya†, Yusuke Kumagae†, Yusuke Ichikawa†, and Hiroshi Sawada†

†NTT Service Evolution Laboratories, NTT Corporation, Japan
{takaya.noriko, kumagae.yusuke, ichikawa.yusuke, sawada.hiroshi}@lab.ntt.co.jp

Abstract - The transfer learning method will be modified more effectively for the item purchase prediction on web marketing. Acquiring a various related site information, it would give more accurate prediction than a single site analysis. These multiple EC sites have two problems that 1) some item purchase data are inconsistent to another data set and indeed lower the prediction accuracy and 2) the item's information of brands, categories, prices, and item names in multiple EC sites are sparse and imbalance. Analyzed these characteristics, we propose an ensemble-based approach that effectively aggregates weak classifiers by efficiently avoiding the negative learning effect. Furthermore, we convert the item information to an abstract form. These methods are validated by the actual purchase logs over several Japanese fashion EC sites.

Keywords: Transfer learning, Marketing, Machine learning, EC

1 INTRODUCTION

Administrator of an EC site has been constructing a model to predict their customers purchase. Knowing the information about user's purchase behaviors in other EC sites, they would get more precise insight. Fig. 1[1] shows that the information of the customer behaviors on multiple site will help more precise analysis of the prediction model. The *Transfer Learning* method is known as one of the effective approach for analyzing these transfer situation. In this method, the target domain for which predictions are to be made is called *Target* and the different domain used for learning is called *Source*. The purpose of transfer learning is to utilize the knowledge acquired from the source to improve prediction performance in the target domain. The negative transfer is pointed as a known problem for adopting this method [1]. Fig. 2 describes this negative transfer where the attributions of target data are different from that of source data. In this case, using the source data in learning phase degrades the accuracy. Rosenstein showed a specific example [1]. While their goal was predicting whether the target person would attend or not a specific meeting, the training data were drawn from two people with different attributions (academic and military). Using this training data decreased prediction accuracy. Naturally, there are target/source pairings which improve or degrade accuracy. Therefore, in transfer learning, it is a problem that how we avoid negative transfer effect and how we find similar data. Our proposal is a purchase-based model to predict item sales. This *OptTrBagg* (Opt Transfer Bagging) model is more tolerant to negative transfer. The algorithm is based primarily on *TrBagg*, which is an extension of bagging method, and efficiently drops inadequate base classifiers in aggregation phase.

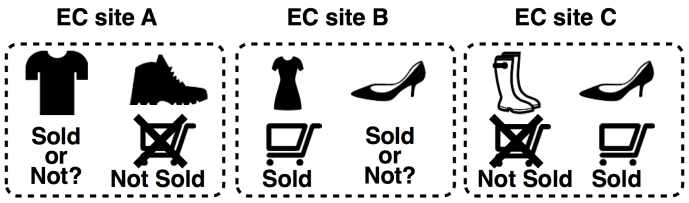


Figure 1: An illustration of our research settings. Traditional item purchase behavior research focuses on single EC site's information (in the above figure, EC site A). Our proposal collects purchase of multiple EC sites (in the above figure, EC sites A, B, and C) and constructs a model to predict whether the item would sell or not.

The inconsistency of each brand or category attribution on the multiple site would cause a identity difficulty for the model. Adopting the abstract description will give the answer to avoid this problem. The effectiveness of our approach is validated by experimentation actual purchase data.

In Section 2, we explain related work on modeling for purchase behavior, ensemble learning, and transfer learning. In Section 3, we introduce TrBagg as the baseline, and propose OptTrBagg, our approach. In Section 4, we explain construction of features across multiple EC sites. In Section 5, we explain our actual purchase information datasets and show the results of experiments. Moreover, in this section, we explain how transfer learning changed the prediction models. Finally, in Section 6, we summarize this paper.

2 RELATED WORK

2.1 Modeling Purchase Behavior

In the area of modeling purchase behavior in e-commerce, there are two approaches, item based prediction and session based prediction. Item based prediction construct models that use the item's own information such as price, category, and item name to predict if the item would be sold. On the other hand, session based prediction construct models that process the user's activity information such as how long the user peruses an item, how many times the user clicked a link, and what queries the user input to predict whether the user will purchase the item in the current session. In item based prediction, Wu and Bolivar discussed the problem of prediction of item purchase behavior [2]. Within eBay[2], which is the

[1]All pictograms used in this paper are from The Noun Project (http://thenounproject.com/). Boots designed by Luis Prado.

[2]http://www.ebay.com/

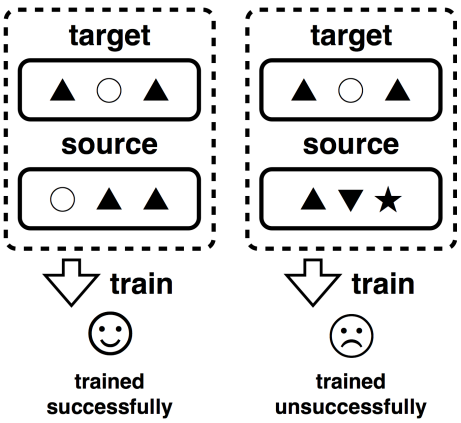


Figure 2: An example of negative transfer. In the left image, the target and source data have similar distribution and model training is successful. In contrast, the right image shows different distributions triggering negative transfer and the failure of model training.

largest Internet auction site, they assigned features to items posted on eBay and predicted the result of purchase by logistic regression. Our research resembles theirs in that it assigns features to each item and prediction of the result, but differs in that is uses information from several EC sites. They also discussed item based prediction but with the goal of predicting item rarity [3]. For session based prediction, Kim uses neural networks that have different hidden layers and aggregate their classification results to predict item purchase behavior in an EC site [4]. Our research resembles this work in that it uses ensemble learning, but differs in that it uses transfer learning. Moe and Fader assign features, such as page transitions and split time period, to the user session on an EC site and predict whether the user will purchase any items in this session or not [5]. Poel and Buckinx also discussed this problem [6]. Guo and Agichtein predict whether the user is now trying to purchase an item or just browsing [7]. They used a Markov chain model and compared the transition probabilities between user purchase and other. In addition, there is similar research in the area of display advertising in websites [8]–[10]. These works used user behavior information extracted from click-through logs as features, whereas we use user behavior-independent information.

Limayem analyzed user purchase behavior on the Internet using factor models [11]. The models they use are based on hypothesis and statistical test between two factors, such as there being a positive relationship between Personal Innovativeness and Intention. Bellman investigated lifestyle and purchase behavior of the user who was called Wired those days [12].

2.2 Ensemble Learning and Transfer Learning

The concept of ensemble learning is to generate several weak learners to reduce variance and improve accuracy. Bagging [13] generates several base classifiers (decision tree is used as base classifier in the original paper) and aggregates their classification results by simply majority voting (in regression problems, values of each weak learner are averaged). Freund proposed AdaBoost [14], which does not aggregate base classifier's results naively. AdaBoost weights each base classifier by empirical error and final prediction is yielded by weighted voting.

Transfer learning is widely used in link prediction [15], displaying advertise [16], object detection in image processing [17], regression [18], video summarization [19], text classification [20]. Kamishima proposed TrBagg [21], which applies bagging to transfer learning (see in Section 3.1). Dai proposed TrAdaBoost which applies AdaBoost to transfer learning [22]. Rosenstein proposed ExpBoost which also applies AdaBoost to transfer learning [23]. Pararoe expanded TrAdaBoost and ExpBoost to cover regression problem [18]. Given that TrBagg offers ease of implementation and tuning, the possibility of parallel computation, and superior accuracy, we propose create our algorithm. Daume proposed a transfer learning method that converts both target and source features simply [24]. For example, the F dimension feature vector $\mathbf{x} \in \mathbb{R}^F$ in target domain $\mathcal{D}_T$ is converted to new feature vector $\Phi^T(\mathbf{x}) =< \mathbf{x}, \mathbf{0}, \mathbf{x} >\in \mathbb{R}^{3F}$, where $\mathbf{0}$ is empty vector $< 0, \cdots, 0 >\in \mathbb{R}^F$. F dimension feature vector $\mathbf{x} \in \mathbb{R}^F$ in source domain $\mathcal{D}_S$ is also converted to new feature vector $\Phi^S(\mathbf{x}) =< \mathbf{x}, \mathbf{x}, \mathbf{0} >$. The converted vectors are used to train a model. We adopt this approach as our baseline in the experiments.

3 ENSEMBLE TRANSFER LEARNING

First, we explain TrBagg as baseline method, and next we propose OptTrBagg algorithm.

3.1 Baseline Method: TrBagg

TrBagg is the extension of bagging, which is proposed by Kamishima [21](Algorithm 1). The inputs are target data $\mathcal{D}_T$, source data $\mathcal{D}_S$, and the number of initial base classifiers N. In training, the output is a set of base classifiers $\mathcal{F}^* = \{\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n\}$. The number of output is n and is not greater than the number of initial base classifiers N. The algorithm is as follows.

First, we generate merged data sets $\mathcal{D} = \mathcal{D}_T \cup \mathcal{D}_S$, the union set of target and source data. We get classifier $\hat{f}_0$ learned from $\mathcal{D}_T$ in step 4. In the iteration of t, we generate training data $\mathcal{D}'_t$ by bootstrap sampling (random sampling which allows duplication) from $\mathcal{D}$ and get base classifier $\hat{f}_t$ learned from $\mathcal{D}'_t$. By repeating this iteration N times, we get the set of base classifiers $\mathcal{F} = \{\hat{f}_0, \hat{f}_1, \cdots, \hat{f}_N\}$.

Next, we filter the base classifiers $\mathcal{F}$ by Algorithm 2. In step 3, we sort $\mathcal{F}$ in ascending order of their empirical errors on the target set $\mathcal{D}_T$. From step 7, we check each base classifier f_t according to the empirical error. That is, we check whether the empirical error of majority voting is improved or not by the addition of f_t on $\mathcal{D}_T$ to the set of base classifiers $\mathcal{F}'$. The result of prediction $\hat{c}$ by majority voting on unknown data $\mathbf{x}$ by the set of models $\mathcal{F}'$ is determined by

$$\hat{c} = \arg\max_{c \in \mathcal{C}} \sum_{\hat{f}_t \in \mathcal{F}'} \mathrm{I}[c = \hat{f}_t(\mathbf{x})], \tag{1}$$

Algorithm 1 TrBagg

1: function **Training**
2: **INPUT** $\mathcal{D}_T$, $\mathcal{D}_S$, N
3: $\mathcal{D} = \mathcal{D}_T \cup \mathcal{D}_S$
4: $\mathcal{F} = \{\hat{f}_0\}$; $\hat{f}_0$: learned from $\mathcal{D}_T$
5: **for all** $t = 1$ to N **do**
6: $\mathcal{D}'_t \leftarrow$ generated by bootstrap from $\mathcal{D}$
7: $\hat{f}_t$: learned from $\mathcal{D}'_t$
8: $\mathcal{F} = \mathcal{F} \cup \hat{f}_t$
9: **end for**
10: $\mathcal{F}^* = \mathbf{Filtering}(\mathcal{F}, \mathcal{D}_\mathbf{T})$
11: **OUTPUT** $\mathcal{F}^* : \{\hat{f}_0, \hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n\}$ $(n \leq N)$

Algorithm 2 Filtering

1: function **Filtering**
2: **INPUT** $\mathcal{F} : \{\hat{f}_0, \hat{f}_1, \hat{f}_2, \cdots, \hat{f}_N\}$, $\mathcal{D}_T$
3: $< f_0, f_1, \cdots, f_N >$: sort $\mathcal{F}$ by empirical error on $\mathcal{D}_T$ in ascending
4: $\mathcal{F}' = \{f_0\}$
5: $\mathcal{F}^* = \{f_0\}$
6: $e \leftarrow$ empirical error of $\hat{f}_0$ on $\mathcal{D}_T$
7: **for all** $t = 1$ to N **do**
8: $\mathcal{F}' = \mathcal{F}' \cup f_t$
9: $e' \leftarrow$ empirical error of majority voting $\mathcal{F}'$ on $\mathcal{D}_T$
10: **if** $e' \leq e$ **then**
11: $\mathcal{F}^* = \mathcal{F}^* \cup \mathcal{F}'$
12: $e' = e$
13: **end if**
14: **end for**
15: **OUTPUT** $\mathcal{F}^* : \{\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n\}$ $(n \leq N)$

where I$[cond]$ is an indicator function that returns 1 if condition $cond$ is true, and $\mathcal{C}$ is the set of classes. If empirical error is improved, we set all $\mathcal{F}'$ to $\mathcal{F}^*$.

Finally, we get the set of base classifiers $\mathcal{F}^*$. The aim of this filtering is to prevent negative transfer, since transfer learning is not always assured to be effective. That is to say, in filtering iteration i, using just the target data may yield higher performance. We propose a method that is more effective in avoiding negative transfer.

3.2 Proposed Method: OptTrBagg

The method is based on the idea of not using source data that degrade prediction accuracy; OptTrBagg overcome this problem by filtering out the base classifiers. Algorithm 3 and Fig. 3 describe the procedure. The difference between OptTrBagg and TrBagg is the learning process involving base classifier $\hat{f}_t$. In iteration t, our approach pays attention to target data $\mathcal{D}'_{t,T}$ which are contained in training data $\mathcal{D}'_T$ (in step 6 and 7). We get the base classifier $\hat{f}_{t,T+S}$ learned from $\mathcal{D}'_t$ and another base classifier $\hat{f}_{t,T}$ learned from $\mathcal{D}'_{t,T}$ (in step 8 and 10). Incidentally $\hat{f}_{t,T+S}$ is denoted as $\hat{f}_t$ in Algorithm 1. Using empirical error on $\mathcal{D}_T$ to comparing $\hat{f}_{t,T+S}$ with $\hat{f}_{t,T}$, we use the model as $\hat{f}_t$ in iteration t (in step 17).

The base classifier $\hat{f}_{t,T+S}$ is learned from both the target and source data because $\mathcal{D}'_t$ contains target and source data. If learning process with source data is effective, the empirical error of $\hat{f}_{t,T+S}$ which is learned from target and source

Algorithm 3 OptTrBagg

1: function **Training**
2: **INPUT** $\mathcal{D}_T$, $\mathcal{D}_S$, N
3: $\mathcal{D} = \mathcal{D}_T \cup \mathcal{D}_S$
4: $\mathcal{F} = \{\hat{f}_0\}$; $\hat{f}_0$: learned from $\mathcal{D}_T$
5: **for all** $t = 1$ to N **do**
6: $\mathcal{D}'_t \leftarrow$ generated bootstrap from $\mathcal{D}$
7: $\mathcal{D}'_{t,T} = \mathcal{D}'_t \cap \mathcal{D}_T$
8: $\hat{f}_{t,T+S}$: learned from $\mathcal{D}'_t$
9: $e_{T+S} \leftarrow$ empirical error of $\hat{f}_{t,T+S}$ on $\mathcal{D}_T$
10: $\hat{f}_{t,T}$: learned from $\mathcal{D}'_{t,T}$
11: $e_T \leftarrow$ empirical error of $\hat{f}_{t,T}$ on $\mathcal{D}_T$
12: **if** $e_T \leq e_{T+S}$ **then**
13: $\hat{f}_t = \hat{f}_{t,T}$
14: **else**
15: $\hat{f}_t = \hat{f}_{t,T+S}$
16: **end if**
17: $\mathcal{F} = \mathcal{F} \cup \hat{f}_t$
18: **end for**
19: $\mathcal{F}^* = \mathbf{Filtering}(\mathcal{F}, \mathcal{D}_\mathbf{T})$
20: **OUTPUT** $\mathcal{F}^* : \{\hat{f}_0, \hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n\}$ $(n \leq N)$

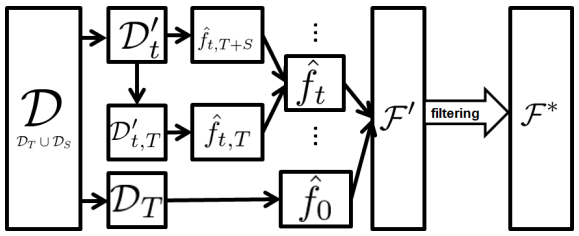


Figure 3: An overview of OptTrBagg. $\mathcal{D}'_t$ is generated by bootstrap sampling from both target and source data $\mathcal{D} = \mathcal{D}_T \cup \mathcal{D}_S$. $\mathcal{D}'_{t,T}$ is extracted from $\mathcal{D}'_t$ by intersection of target data $\mathcal{D}_T$. Classifier $\hat{f}_t$ is selected from $\hat{f}_{t,T+S}$ learned from $\mathcal{D}'_t$ and $\hat{f}_{t,T}$ learned from $\mathcal{D}'_{t,T}$ by its empirical error.

data may be smaller than the empirical error of $\hat{f}_{t,T}$ which is learned from target data. In this case, $\hat{f}_{t,T+S}$ is adopted as $\hat{f}_t$ and this result is equal to TrBagg's process. On the contrary, if learning by source data fails, the empirical error of $\hat{f}_{t,T+S}$ may be larger than empirical error of $\hat{f}_{t,T}$. In this case, OptTrBagg adopt $\hat{f}_{t,T}$ as $\hat{f}_t$.

In eliminating negative transfer, both TrBagg and OptTrBagg filter base classifiers by majority voting. In addition to this filtering, OptTrBagg checks whether each base classifier degrades accuracy or not. Hence, OptTrBagg more efficiently the negative transfer classifiers that can slip into aggregation of base classifiers. That is to say, OptTrBagg can be interpreted as extension of TrBagg where source data that degrade accuracy are removed.

3.3 Difference Between OptTrBagg and TrBagg Viewing from Bagging

We explained OptTrBagg as an extension of the TrBagg algorithm in the previous section, but OptTrbagg can also be interpreted as an extension of bagging [13]. In iteration t, bagging generates training data $\mathcal{D}'_t$, and base classifier $\hat{f}_t$ learned from $\mathcal{D}'_t$. Finally, bagging gets the set of base classi-

Table 1: Definition of symbols used in constructing features.

Symbol	Definition
item_i	A fashion item.
r_i	Price of item_i.
c_i	Category of item_i (e.g. Polo shirt, Denim jeans, and Scarf).
b_i	Brand of item_i (e.g. LOUIS VUITTON[3] and Burberry[4]).
t_i	Item name of item_i (e.g. *"Oxford Button-Down Shirt"*).
s_i	Boolean indicating whether item_i was sold (= 1) or not (= 0).
$\mathcal{I}$	Set of all items.
$\mathcal{I}_{c,i}$	Set of items whose category is c_i, $\{\forall \text{item}_j \in \mathcal{I}, c_j = c_i\}$.
$\mathcal{I}_{b,i}$	Set of items whose brand is b_i, $\{\forall \text{item}_j \in \mathcal{I}, b_j = b_i\}$.
$\mathcal{I}_s$	Set of items sold, $\{\forall \text{item}_i \in \mathcal{I}, s_i = 1\}$.
$\mathcal{I}_{K,i}$	Top K items that have similar item name to item_i.
$\lvert\mathcal{I}_*\rvert$	Size of set of items $\mathcal{I}_*$ (* indicates some conditions).
$\langle r_i \rangle_{i \in \mathcal{I}_*}$	Average price of a set of items $\mathcal{I}_*$ (* indicates some conditions).

fiers $\mathcal{F}^* = \{\hat{f}_1, \cdots, \hat{f}_N\}$.

Both algorithms, bagging and OptTrBagg, generate training data $\mathcal{D}'_{t,T}$ from $\mathcal{D}_T$ by bootstrap and get base classifier $\hat{f}_{t,T}$. Although bagging uses $\hat{f}_{t,T}$ as $\hat{f}_t$, OptTrBagg decides $\hat{f}_t$ by comparing $\hat{f}_{t,T}$ with $\hat{f}_{t,T+S}$, which is learned from the data bootstrapped from source data $\mathcal{D}_S$ and $\mathcal{D}_T$. As explained in Section 3.2, if source data are effective in the learning phase, $\hat{f}_{t,T+S}$ is adopted as $\hat{f}_t$. On the contrary, if source data may cause negative transfer, $\hat{f}_{t,T}$ is adopted as $\hat{f}_t$ and this result is equivalent to that of bagging.

That is to say, OptTrBagg is interpreted as extension of bagging to transfer learning where source data that can improves accuracy is used.

4 FEATURES ACROSS MULTIPLE EC SITES

Constructed a model to predict the selling, eight features were proposed based on the purchase prediction model of Wu and Bolivar [2]. There are two kinds of features, six attribution are based on item attributes of item's price, category and brand, and two name based features were constructed by item name. The symbols used in explaining features are defined in Table 1.

4.1 Attribution Based Features

Attribution based features are constructed from the information about price of sold items. The detail is as follows;

- We cannot use the price of item_i, r_i as a feature directly because it differs by brand and/or category of item_i. Comparing the prices of different categories, such as underwear and suit priced, is nonsense. It is important to consider the item price as the different from an average price.
- The popularity of a brand or category on each site differ. Therefore, the direct comparison of these attributions is not fair. These information will be transform the abstract form.

The proposal methods are;

- **Category Averaged Price** : $r_i - \langle r_i \rangle_{i \in \mathcal{I}_{c,i}}$, difference between price of item_i, r_i and average price of category c_i.
- **Category Averaged Sold Price** : $r_i - \langle r_i \rangle_{i \in (\mathcal{I}_{c,i} \cap \mathcal{I}_s)}$, difference between price of item_i, r_i and average price of sold items in category c_i.
- **Category Hotness** : $\frac{|\mathcal{I}_{c,i} \cap \mathcal{I}_s|}{|\mathcal{I}_{c,i}|}$, the selling rate of items whose category is c_i.
- **Brand Averaged Price** : $r_i - \langle r_i \rangle_{i \in \mathcal{I}_{b,i}}$, difference between price of item_i, r_i and average price of brand items b_i.
- **Brand Averaged Sold Price** : $r_i - \langle r_i \rangle_{i \in (\mathcal{I}_{b,i} \cap \mathcal{I}_s)}$, difference between price of item_i, r_i and average price of sold items whose brand is b_i.
- **Brand Hotness** : $\frac{|\mathcal{I}_{b,i} \cap \mathcal{I}_s|}{|\mathcal{I}_{b,i}|}$, the selling rate of items whose brand is b_i.

4.2 Name Based Features

The name based features is follows;

- The price, category, and brand information annotated in item captures item purchase tendency, it does not contain other information such as color, shape, and feel. For example, for the item named "Cute Mori-Girl[5] style! Over knee high socks with Natural color made by Paralleled Yarn" existing features can represent only the category "knee high". However, using name based features allows the learning phase to refer to attributes that directly impact the user's sense of fashion and preference, such as "Mori-Girl, Natural color, and paralleled yarn."
- The item name causes sometimes misunderstanding because of the sparsity problem and item name brevity. Similar to category and brand, we have to convert item name information into abstract form to be able to use it as features.
- For abstracting item name information, we adopt the hypothesis that similar items have similar purchase tendencies. To calculate item similarity, we regard item name as a set of characters (e.g. we regard "Oxford Button-Down Shirt" as {o, x, f, r, d, ' ', b, u, t, n, '-', w, s, h, i}) and we employ the Jaccard coefficient to measure the similarity of the names of two items.

Based on these assumptions, we construct name based features.

- **Name Averaged Sold Price** : $r_i - \langle r_i \rangle_{i \in (\mathcal{I}_{K,i} \cap \mathcal{I}_s)}$, difference between price of item_i, r_i and average price of items with similar top K item names.
- **Name Hotness** : $\frac{|\mathcal{I}_{K,i} \cap \mathcal{I}_s|}{K}$, the selling rate of items with similar top K item names.

[3] http://www.louisvuitton.com
[4] http://www.burberry.com
[5] "Mori-Girl" is a Japanese fashion trend for young women invoking a soft, forest-like tone.

5 EXPERIMENTS: PURCHASE PREDICTION

In this experiment, we construct model that predicts of the sales. Our model takes, as inputs, the attributes of the item of price, category, brand, and item name, and its output is binary value indicating the sales results. First, we explain our actual purchase dataset gathered from multiple EC sites and how we collected it.

5.1 Multiple EC Site DataSet And the Crawling Scheme

Our project was working with about 1,000 EC site users (called "Panel") and collects online purchase behavior activity logs over a long period of time by the log crawling software (called "Client"). Fig. 4 shows our data collecting system. User behavior logs collected by Client were annotated by Support Vector Machine [25] based model, and converted into certain format that was suitable for analysis. We use this purchase information in the following experiments. The main purpose of this experiment is making the prediction on each customer. The dataset was collected over the 23 Japanese EC sites listed in Table 2. We made sampling in order to ensure balance in terms of positive/negative data. For the EC sites of MUJI, Amazon, and RAKUTEN, we filtered out unrelated data to fashion items.

We performed two experiments. In Experiment 1 (Section 5.2), we used all the 23 EC sites. In Experiment 2 (Section 5.3), we selected 8 EC sites, 4 for target EC sites and the other 4 for source EC sites.

First, we explain about each EC site which were used as target data in Experiment 2. OUTLET PEAK is a fashion EC site and lays in a stock of items from fashion brands directly. MUJI is the largest commodity brand in Japan. Their EC site handles only one brand, "No Brand Quality Goods". Nissen originally handles mail order. Now they handle various fashion items such as woman's shirt, men's jacket, and baby's pajamas. By contrast, PEACH JOHN handles mainly woman's underwear with their own brand, "PJ".

Next, we explain about each EC site which were used as source data in Experiment 2. GLAMOUR SALES is the EC site focused on deal of the day, their sales have 157 hours limitation. They handle over 1,200 brands. ZOZOTOWN is the largest fashion EC site of Japan. They handle over 2,000 brands and over 130,000 items. They also manage a social networking service, ZOZOPEOPLE. UNIQLO is not only an EC site but also the largest casual fashion brand in Japan, such as MUJI. Their EC site handles very few brands; UNIQLO (their main brand), g.u (lower price items), and UT (tee-shirts only). RAKUTEN is a kind of online shopping mall. They are largest EC site of the business type called B2B2C (Business to Business to Consumer) in Japan. They do not deal with consumers directly (Business to Consumer), but also provide an E-Commerce platform where other companies are able to build up their own EC sites (called mall) in RAKUTEN. By opening their own mall on RAKUTEN, companies deal with consumers directly. Currently RAKUTEN has about 40,000 malls and various items over 0.1 billion from fashion clothes

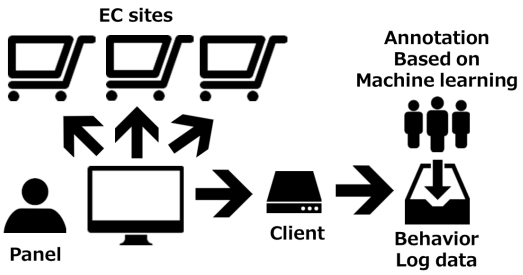


Figure 4: Data crawling scheme used by the project. Client software installed on Panel member's personal computers captures information about purchase behavior such as items that the Panel bought or queries they entered in Google, and sends them to our server. These data are converted into format suitable for analysis by statistical methods. After format conversion, the information is annotated by Support Vector Machine based method and checked by humans.

to real estate. Thus, there are many brands and categories. These EC sites on our experiment have several characteristics depending on their origin and business model.

5.2 Experiment 1: Single Source Settings

5.2.1 Parameter Settings

For Experiment 1, we prepared to set 3 parameters; the number of initial base classifiers N, top K size using name based features, and the data size of bootstrap $|\mathcal{D}'_t|$. The number of initial base classifiers N was fixed to 100. In name based features, the top $K = 10$ items were used to identify similar items. The data size of each bootstrap $|\mathcal{D}'_t|$ equaled source data size, $|\mathcal{D}_S|$, 17,398.

We selected standard bagging [13], Frustratingly Easy Domain Adaptation [24], and TrBagg [21] as our verification methods. The base classifier used in each algorithm was C5.0 [26] decision tree, which is an extension of the C4.5 [27] algorithm. Abbreviations of method names are defined in Table 3. We performed a five-fold cross-validation test and used the average values.

5.2.2 Results

We tested whether the proposed method was superior to other methods in terms of accuracy. In experiment 1, we used all EC sites except RAKUTEN as target and RAKUTEN as source. The results are shown in Table 4. In Table 4, the columns list the target EC sites. Each row lists, from left to right, target EC site's name, the number of items in the site, and the remaining cells show the prediction accuracy for all methods. The values are the average of output by the five-hold cross-validation test.

We assessed these results from two view points;

1. whether OptTrBagg was superior to TrBagg

2. whether transfer learning worked effectively in item purchase prediction

Table 2: A list of target EC sites and their size of items.

Site Name	URL	# of items
FLAG SHOP	flagshop.jp	116
0101	0101.jp	124
SELECT SQUARE	selectsquare.com	128
Wacoal	wacoal.jp	146
SELESONIC	selesonic.com	156
SHOP CHANNEL	shopch.jp	220
ELLE SHOP	elleshop.jp	222
fashionwalker.com	fashionwalker.com	242
i LUMINE	i.lumine.jp	282
YOOX	yoox.com/jp	284
WORLD ONLINE	store.world.co.jp	300
OUTLET PEAK	outletpeak.com	322
MAGASEEK	magaseek.com	358
MUJI	muji.net/store	384
BRANDELI	brandeli.com	524
Nissen	nissen.co.jp	556
GILT	gilt.jp	584
Javari	javari.jp	676
PEACH JOHN	peachjohn.co.jp	712
Amazon	amazon.co.jp	988
GLAMOUR SALES	glamour-sales.com	1,382
ZOZOTOWN	zozo.jp	2,130
UNIQLO	uniqlo.com/jp	3,338
RAKUTEN	rakuten.co.jp	17,398

Table 3: Definition of abbreviations of method names.

Abbreviation	Method Name
DT	C5.0 decision tree [26]
BG	bagging [13]
FRUST	Frustratingly Easy Domain Adaptation [24]
TB	TrBagg [21]
OPT	OptTrBagg (proposed)

First, OptTrBagg showed better performance than TrBagg in all target data. In predicting MAGASEEK, OptTrBagg outperformed TrBagg as 5.9 points. Compared to Frustratingly Easy Domain Adaptation, OptTrBagg was superior for 22 of the 23 data sets. This confirms the validity of our OptTrBagg.

Second, we compare OptTrBagg to the standard learning methods. In comparison with standard learning methods, bagging outperformed standard C5.0 decision tree. This indicates the effectiveness of ensemble learning. Comparing OptTrBagg to bagging, OptTrBagg showed superiority to bagging in 12 EC sites. This means that prediction results by transfer learning were effective in some situations, but were not in 12 EC sites. We tried to find some tendencies when transfer learning did not work effectively.

Fig. 5 shows the tendency between the number of item size for each EC site and the improvement of accuracy achieved by OptTrBagg. X axis indicates the number of items in each EC sites and Y axis indicates the increase of accuracy offered by transfer learning (improved score between of OptTrBagg accuracy and bagging accuracy). This figure shows that the effectiveness of transfer learning. EC sites with over 1,500 items, such as ZOZO and UNIQLO, have sufficient items for constructing the prediction model. Transfer learning improved the accuracies of EC sites which have less than 200 items. This result indicates that these EC sites have insufficient data to construct a prediction model. Transfer learning effectively worked to train a model in these situations. On the other hand, the accuracies of transfer learning were inferior of bagging in some EC sites such as MUJI (-3.131 points) and OUTLET PEAK (-2.481 points). Next experiment was conducted to determine the most appropriate pairing and to examine the prediction results.

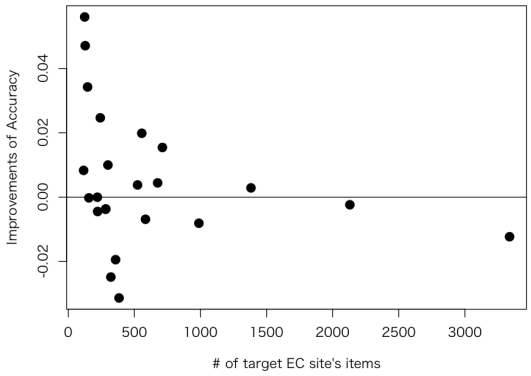


Figure 5: A scatter plot of the item size of target EC sites (X axis) and the accuracy improvements offered by OptTrBagg (Y axis).

5.3 Experiment 2: Multiple Source Settings

In Experiment 2, we intended to identify appropriate target/source pairs that improve accuracy. Then we checked the similarity of price distribution between source EC site and target EC sites in this experiment:

5.3.1 Parameter Settings

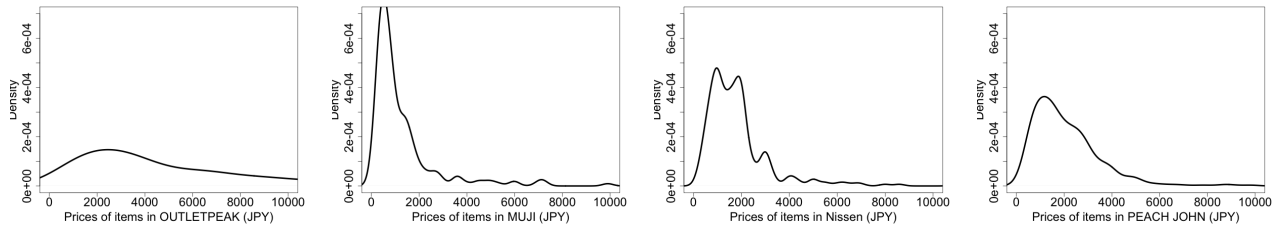
For Experiment 2, we prepared to set 3 parameters; the number of initial base classifier N, the top K size using name based features, and the data size of bootstrap $|\mathcal{D}'_t|$. At first, the number of initial base classifiers N was fixed to 100. In name based features, the top $K = 10$ items was used to identify similar items. The data size of each bootstrap $|\mathcal{D}'_t|$ equaled to the size of each source data size $|\mathcal{D}_S|$. The base classifier used each algorithm was C5.0 decision tree.

We selected OUTLET PEAK, MUJI, Nissen, and PEACH JOHN as target EC sites having the number of items around 500, because the prediction accuracies of OUTLETPEAK and MUJI were decreased by transfer learning, and Nissen and PEACH JOHN were increased by transfer learning. As source data in addition to RAKUTEN, we added 3 sites; GLAMOUR SALES, ZOZOTOWN, and UNIQLO, having the number of records around 1000.

Fig. 6 shows the price density distribution of items in each target EC site, OUTLET PEAK, MUJI, Nissen, and PEACH JOHN. Fig. 7 shows the price density distribution of items in each source EC site, GRAMOUR SALES, ZOZOTOWN, UNIQLO, and RAKUTEN. Fig. 6 and Fig. 7 shows difference of price distributions in each EC site clearly. If the prediction performance depended on the similarity of features associated with price, the transfer learning with similar price distribution would improve the accuracy.

Table 4: Results of experiment 1. Values are average accuracy of five-hold cross-validation. Bold number indicates the best accuracy among all learning methods and italic number indicates the best accuracy among transfer learning methods.

		Standard Learning		Transfer Learning		
target	# of items	DT	BG	FRUST	TB	OPT
FLAG SHOP	116	0.8442	0.9221	0.9047	0.9047	***0.9304***
0101	124	0.9033	0.9357	0.9837	***0.9917***	***0.9917***
SELECT SQUARE	128	0.8043	0.8988	0.8677	0.8911	***0.9458***
Wacoal	146	0.7945	0.8149	0.8147	0.8428	***0.8492***
SELESONIC	156	0.7567	**0.7823**	0.7052	0.7498	*0.7821*
SHOP CHANNEL	220	0.7409	**0.7818**	0.7409	0.7682	***0.7818***
ELLESHOP	222	0.7524	**0.8157**	0.7567	0.7747	*0.8112*
fashionwalker.com	242	0.9340	0.9547	0.9710	0.9628	***0.9793***
i LUMINE	282	0.7555	**0.7768**	0.7410	0.7236	*0.7731*
YOOX	284	0.9543	**0.9648**	0.9541	0.9506	*0.9612*
WORLD ONLINE	330	0.7700	0.8033	0.7700	0.8100	***0.8133***
OUTLET PEAK	322	0.7484	**0.8042**	0.7607	0.7517	*0.7794*
MAGASEEK	358	0.7655	**0.8210**	0.7375	0.7431	*0.8016*
MUJI	384	0.8098	**0.8358**	0.7810	0.7940	*0.8045*
BRANDELI	542	0.8016	0.8149	0.7825	0.8112	***0.8188***
Nissen	556	0.7428	0.7769	0.7356	0.7788	***0.7968***
GILT	584	0.7757	**0.8049**	0.7912	0.7364	*0.7981*
Javari	676	0.9275	0.9512	***0.9556***	0.9542	***0.9556***
PEACH JOHN	712	0.6756	0.6699	0.6489	0.6517	***0.6854***
Amazon	988	0.8290	**0.8320**	0.8057	0.8229	*0.8239*
GLAMOUR SALES	1,382	0.7771	0.7916	0.7663	0.7728	***0.7945***
ZOZOTOWN	2,130	0.9915	**0.9930**	0.9901	0.9901	*0.9906*
UNIQLO	3,338	0.6690	**0.6773**	*0.6699*	0.6606	0.6651
# of best accuracy among transfer learning		-	-	2	1	22
# of best accuracy among all methods		0	12	1	1	12



(a) (b) (c) (d)

Figure 6: Price distribution of items in each target EC site used in Experiment 2, OUTLET PEAK (Fig. 6(a)), MUJI (Fig. 6(b)), Nissen (Fig. 6(c)), and PEACH JOHN (Fig. 6(d)).

5.3.2 Accuracy of Target / Source Pair and Their Price Distribution

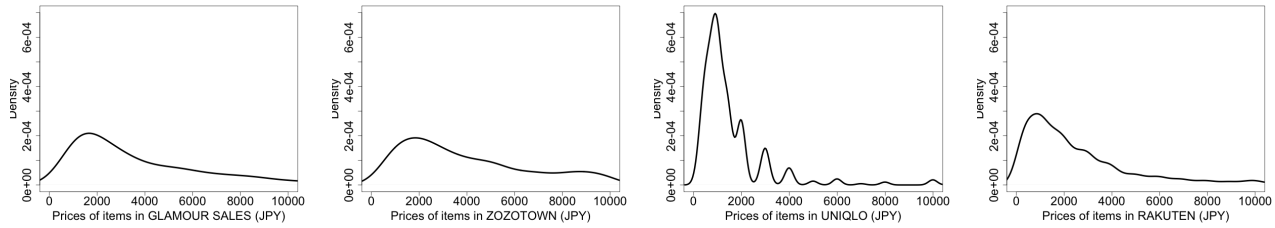
In Table 5, abbreviations of method names are defined. The results are shown in Table 5. The values in each cell were averaged accuracy of the five-hold crossvalidation. The **bold** number indicates the best accuracy among all learning methods and the *italic* number indicates the best accuracy among transfer learning methods.

First, the improvement of accuracy does depend on the target/source pairing. In OUTLET PEAK and MUJI, which transfer learning failed to predict in Experiment 1, transfer learning yielded better accuracy than standard bagging. Overall, none of the source data sets yielded the best accuracy for all targets (*silver bullet*) and none of the source data sets yielded the worst accuracy for all targets. It indicates the importance for transfer learning to select source data when constructing prediction models.

Second, we focused on the similarity of price distribution (Fig. 6 and Fig. 7) of each EC site. In each target and source pairing which yield best accuracy, the price distribution of the source EC site was similar with the target EC site. For example, the price distribution of OUTLET PEAK was similar with that of GLAMOUR SALES. These two price distribution of MUJI and UNIQLO ware also skewed. These observations suggest that the validity of transfer learning is determined by the similarity of features between the source EC data and the target EC data.

6 CONCLUSION

In this paper, we focused on prediction of the sales results using multiple EC site's purchase information. In order to construct the effective model, we converted the item's information such as brand, category, price, and item name into suitable formulation. We also intend to develop the effective method for finding the optimal pair of target and source data sets in transfer learning. The proposed OptTrbagg was a new method adopting transfer learning on EC marketing. We examined many Target and Source pairing and confirmed supe-



(a) (b) (c) (d)

Figure 7: Price distribution of items in each source EC site used in Experiment 2, GRAMOUR SALES (Fig. 7(a)), ZOZOTOWN (Fig. 7(b)), UNIQLO (Fig. 7(c)), and RAKUTEN (Fig. 7(d)).

Table 5: Results of experiment 2. Values are average accuracy of five-hold cross-validation. Bold number indicates the best accuracy among all learning methods and italic number indicates the best accuracy among transfer learning methods. In each row, bagging is standard learning method which uses target EC site's information only, and others use source EC site's information.

target	BG	source	FRUST	TB	OPT
OUTLET PEAK	0.8042	GLAMOUR SALES	0.7984	***0.8232***	0.8200
		ZOZOTOWN	0.7577	0.7794	0.7950
		UNIQLO	0.7640	0.8075	0.8104
		RAKUTEN	0.7607	0.7517	0.7794
MUJI	0.8358	GLAMOUR SALES	0.8047	***0.8463***	0.8411
		ZOZOTOWN	0.7865	0.8099	0.8334
		UNIQLO	0.8307	0.8150	***0.8463***
		RAKUTEN	0.7810	0.7940	0.8045
Nissen	0.7769	GLAMOUR SALES	0.7464	0.7716	0.7698
		ZOZOTOWN	0.7375	0.7662	0.7824
		UNIQLO	0.7534	0.7732	0.7606
		RAKUTEN	0.7356	0.7788	***0.7968***
PEACH JOHN	0.6699	GLAMOUR SALES	0.6742	0.6798	0.6741
		ZOZOTOWN	0.6798	0.6742	0.6699
		UNIQLO	0.6784	0.6811	***0.6867***
		RAKUTEN	0.6489	0.6517	0.6854

riority of our method . Experiments on the actual EC site data indicated that OptTrBagg outperformed TrBagg or the other transfer learning methods. Our composition was more tolerant against negative transfer by exploiting the sparsity structures of features of item. OptTrBagg could contribute to find most appropriate pairs of EC sites to determine the optimal pairing for transfer learning.

REFERENCES

[1] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in NIPS'05 Workshop, Inductive Transfer: 10 Years Later (2005).

[2] X. Wu and A. Bolivar, "Predicting the conversion probability for items on c2c ecommerce sites," in Proceedings of the 18th ACM conference on Information and knowledge management, ser. CIKM'09. ACM, pp. 1377–1386 (2009).

[3] D. Shen, X. Wu, and A. Bolivar, "Rare item detection in e-commerce site," in Proceedings of the 18th international conference on World wide web, ser. WWW'09. ACM, pp. 1099–1100 (2009).

[4] E. Kim, W. Kim, and Y. Lee, "Combination of multiple classifiers for the customer's purchase behavior prediction," Decis. Support Syst., vol. 34, pp. 167–175 (2003).

[5] W. W. Moe and P. S. Fader, "Dynamic conversion behavior at ecommerce sites," Manage. Sci., vol. 50, no. 3, pp. 326–335 (2004).

[6] D. V. D. Poel and W. Buckinx, "Predicting online-purchasing behaviour," European Journal of Operational Research, vol. 166, pp. 557–575 (2005).

[7] Q. Guo and E. Agichtein, "Ready to buy or just browsing?: detecting web searcher goals from interaction data," in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR'10. New York, NY, USA: ACM, pp. 130–137 (2010).

[8] S. Pandey, M. Aly, A. Bagherjeiran, A. Hatch, P. Ciccolo, A. Ratnaparkhi, and M. Zinkevich, "Learning to target: what works for behavioral targeting," in Proceedings of the 20th ACM international conference on Information and knowledge management, ser. CIKM'11. New York, NY, USA: ACM, pp. 1805–1814 (2011).

[9] A. Bagherjeiran, A. Hatch, A. Ratnaparkhi, and R. Parekh, "Large-scale customized models for advertisers," in Proceedings of the 2010 IEEE International Conference on Data Mining Workshops, ser. ICDMW'10. Washington, DC, USA: IEEE Computer Society, pp. 1029–1036 (2010).

[10] J. Li, P. Zhang, Y. Cao, P. Liu, and L. Guo, "Efficient behavior targeting using svm ensemble indexing," in ICDM, M. J. Zaki, A. Siebes, J. X. Yu, B. Goethals, G. I. Webb, and X. Wu, Eds. IEEE Computer Society, pp. 409–418 (2012).

[11] M. Limayem, M. Khalifa, and F. Anissa, "What makes consumers buy from internet? a longitudinal study of online shopping," IEEE Transactions on Systems, Man, and Cybernetics, vol. 30, pp. 421–432 (2000).

[12] S. Bellman, G. L. Lohse, and E. J. Johnson, "Predictors of online buying behavior," Commun. ACM, vol. 42, pp. 32–38, (1999).

[13] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, pp. 123–140, (1996).

[14] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in Thirteenth International Conference on Machine Learning. Morgan Kaufmann, pp. 148–156 (1996).

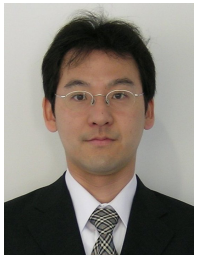
[15] B. Cao, N. N. Liu, and Q. Yang, "Transfer learning for collective link prediction in multiple heterogenous domains," in Proceedings of the 27th International Conference on Machine Learning, ser. ICML'10, J. Fürnkranz and T. Joachims, Eds. Omnipress, pp. 159–166 (2010).
[16] T. Chen, J. Yan, G. Xue, and Z. Chen, "Transfer learning for behavioral targeting," in Proceedings of the 19th international conference on World wide web, ser. WWW'10. New York, NY, USA: ACM, pp. 1077–1078 (2010).
[17] P. Huang, G. Wang, and S. Qin, "Boosting for transfer learning from multiple data sources," Pattern Recogn. Lett., vol. 33, no. 5, pp. 568– 579, (2012).
[18] D. Pardoe and P. Stone, "Boosting for regression transfer," in Proceedings of the 27th international conference on Machine learning, ser. ICML '10, J. Frnkranz and T. Joachims, Eds. Omnipress, pp. 863–870 (2010).
[19] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu, "Video summarization via transferrable structured learning," in Proceedings of the 20th international conference on World wide web, ser. WWW'11. ACM, pp. 287–296 (2011).
[20] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged plsa for cross-domain text classification," in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR'08. ACM, pp. 627–634 (2008).
[21] T. Kamishima, M. Hamasaki, and S. Akaho, "Trbagg: A simple transfer learning method and its application to personalization in collaborative tagging," in Proceedings of the 2009 Ninth IEEE International Conference on Data Mining, ser. ICDM'09, pp. 219–228 (2009).
[22] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in Proceedings of the 24th international conference on Machine learning, ser. ICML'07. ACM, pp. 193–200 (2007).
[23] A. Rettinger, M. Zinkevich, and M. Bowling, "Boosting expert ensembles for rapid concept recall," in Proceedings of the 21st national conference on Artificial intelligence - Volume 1, ser. AAAI'06. AAAI Press, pp. 464–469 (2006).
[24] H. Daume III, "Frustratingly easy domain adaptation," in Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. Association for Computational Linguistics, pp. 256–263 (2007).
[25] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Min. Knowl. Discov., vol. 2, no. 2, pp. 121–167 (1998).
[26] J. R. Quinlan, "C5.0," http://mloss.org/software/view/329/ (2011).
[27] R. Quinlan, "C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)," 1st ed. Morgan Kaufmann, (1992).

(Received November 20, 2014)
(Revised March 10, 2015)

Noriko Takaya Senior Research Engineer, Supervisor, Co-innovation Promotion Project, NTT Service Evolution Laboratories. She received the B.A. degree in in English and American Literature from the Gakushuin University in 1987. In 1995, she joined NTT Advertising, Inc. Since 2010, she has been in NTT Service Evolution Laboratories. She is a member of Information Processing Society of Japan, Japan Marketing Academy and Japan Institute of Marketing Science.

Yusuke Kumagae Yusuke Kumagae received his B.E. and M.E. degrees from University of Tsukuba in 2009 and 2011, respectively. He now works at NTT Service Evolution Laboratories. His research interests include machine learning and consumer behavior.

Yusuke Ichikawa Senior Research Engineer, 2020 Epoch–making Project, NTT Service Evolution Laboratories. He received the B.E. and M.E. degrees in measurement engineering from Keio University, Kanagawa, in 1995 and 1997, respectively. He joined NTT Multimedia Network Laboratories in 1997. Since 1998, he has been engaged in R&D of recommendation systems. He is a senior member of the Information Processing Society of Japan (IPSJ) and received the 2005 IPSJ Yamashita SIG Research Award.



Hiroshi Sawada Hiroshi Sawada received the B.E., M.E. and Ph.D. degrees in information science from Kyoto University, Kyoto, Japan, in 1991, 1993 and 2001, respectively. He joined NTT Corporation in 1993. From 2009 to 2013, he was the group leader of Learning and Intelligent Systems Research Group at the NTT Communication Science Laboratories, Kyoto, Japan. He is now a senior research engineer, supervisor at the NTT Service Evolution Laboratories, Yokosuka, Japan. His research interests include statistical signal processing, audio source separation, array signal processing, machine learning, latent variable model, graph-based data structure, and computer architecture.