IEICE TRANS. INF. & SYST., VOL.E103–D, NO.2 FEBRUARY 2020

PAPER

Users' Preference Prediction of Real Estate Properties Based on Floor Plan Analysis

Naoki KATO[†a)], ***Nonmember***, **Toshihiko YAMASAKI**[†b)], ***Senior Member***, **Kiyoharu AIZAWA**[†c)], ***Fellow***, ***and*** **Takemi OHAMA**[††], ***Nonmember***

SUMMARY With the recent advances in e-commerce, it has become important to recommend not only mass-produced daily items, such as books, but also items that are not mass-produced. In this study, we present an algorithm for real estate recommendations. Automatic property recommendations are a highly difficult task because no identical properties exist in the world, occupied properties cannot be recommended, and users rent or buy properties only a few times in their lives. For the first step of property recommendation, we predict users' preferences for properties by combining content-based filtering and Multi-Layer Perceptron (MLP). In the MLP, we use not only attribute data of users and properties, but also deep features extracted from property floor plan images. As a result, we successfully predict users' preference with a Matthews Correlation Coefficient (MCC) of 0.166.

key words: *floor plan, machine learning, prediction, preference, real estate*

1. Introduction

With the expansion of online services in recent years, e-commerce users have correspondingly increased. Goods are often recommended on e-commerce websites, and its accuracy has improved owing to the recent growth in data. There are two major types of recommender systems: content-based filtering and collaborative filtering, proposed by Goldberg et al. [1]. However, recommender systems are usually only effective for mass-produced items. When almost every item in a category is unique, such as real estate properties, automatic recommendation is difficult.

Under these circumstances, the real estate technology known as Real Estate Tech (RETech) has rapidly become more popular. Moreover, the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) in Japan has conducted social experiments in order to deregulate activities on the Internet in the real estate industry[*]. Therefore, property recommendation on websites has become an important research focus.

Our study aims to implement a recommender system for special data (i.e., real estate properties), which cannot be handled by the general recommender systems described above. For the first step of property recommendation, we predict users' preference for properties using a dataset that includes users' evaluations of properties obtained from Ietty[**], a rental company for real estate properties. Generally, users search for desired properties on property portal websites and contact the real estate companies that list the properties. However, Ietty recommends properties to users on its website, and users evaluate the properties online. Following the evaluation, users can view the properties or rent them. Thus, if an effective method is available to predict users' preference for properties using their evaluation data, automatic property recommendation can be performed more easily.

Accordingly, we proposed a prediction system combining content-based filtering and Multi-Layer Perceptron (MLP) to predict users' property preferences. Moreover, we used deep features of floor plans as the input of the MLP to improve accuracy. Consequently, we succeeded in predicting users' property preferences with a Matthews Correlation Coefficient (MCC) [2] of 0.166.

This study is based on [3], and we add some concrete examples and more thorough discussion. The remainder of this study is organized as follows. In Sect. 2, we discuss the related work on recommendation systems and floor plan image analysis. In Sect. 3, we present the dataset and results of preliminary experiments. In Sect. 4, we describe our proposed method in detail. In Sect. 5, we present the metrics of the main experiments, results, and discussion. In Sect. 6, we present our conclusions.

2. Related Work

2.1 Recommendation Systems

Collaborative filtering is classified into two types: memory-based collaborative filtering, proposed by Goldberg et al. [1], and model-based collaborative filtering, proposed by Breeze et al. [4]. In memory-based collaborative filtering, the users' purchase/evaluation data of items are stored in memory, and collaborative filtering is performed using the stored data each time a recommendation is required. On the other hand, model-based collaborative filtering develops a model that predicts items with high probability of being purchased/evaluated by the users, and collaborative filtering

Manuscript received May 29, 2019.
Manuscript revised October 8, 2019.
Manuscript publicized November 20, 2019.

[†]The authors are with The University of Tokyo, Tokyo, 113–8656 Japan.
[††]The author is with Ietty Co., Ltd., Tokyo, 150–0013 Japan.
a) E-mail: kato@hal.t.u-tokyo.ac.jp
b) E-mail: yamasaki@hal.t.u-tokyo.ac.jp
c) E-mail: aizawa@hal.t.u-tokyo.ac.jp

DOI: 10.1587/transinf.2019EDP7146

[*]http://www.mlit.go.jp/totikensangyo/const/sosei_const_tk3_000120.html (accessed Sep/20/2019, in Japanese)
[**]https://ietty.me/

is performed using the model when recommendation is required. In addition, Pennock [5] proposed hybrid collaborative filtering by combining these two methods, and Xue [6] later improved the performance of hybrid collaborative filtering by data interpolation using clustering.

In the Netflix Prize competition†, Simon [7] greatly improved the performance of recommender systems by using a matrix factorization algorithm, in which both the user factor vector $\boldsymbol{p}_u$ and the item factor vector $\boldsymbol{q}_i$ satisfy (1).

$$\min_{\boldsymbol{q}_i, \boldsymbol{p}_u} \sum_{(u,i)\in\kappa} (r_{ui} - \boldsymbol{q}_i^T \boldsymbol{p}_u)^2 + \lambda(\|\boldsymbol{q}_i\|^2 + \|\boldsymbol{p}_u\|^2) \tag{1}$$

The set κ contains every pair (u, i) that consists of the user u and the item i whose evaluation value r_{ui} exists in the training set. The predicted evaluation value $\hat{r_{ui}} = \boldsymbol{q}_i^T \boldsymbol{p}_u$ is then calculated in the test set. Simon [7] solved this optimization problem using Stochastic Gradient Descent (SGD) optimization with a learning rate γ by iterating (2)–(4).

$$e_{ui} = r_{ui} - \boldsymbol{q}_i^T \boldsymbol{p}_u \tag{2}$$

$$\boldsymbol{q}_i = \boldsymbol{q}_i + \gamma \cdot (e_{ui} \cdot \boldsymbol{p}_u - \lambda \cdot \boldsymbol{q}_i) \tag{3}$$

$$\boldsymbol{p}_u = \boldsymbol{p}_u + \gamma \cdot (e_{ui} \cdot \boldsymbol{q}_i - \lambda \cdot \boldsymbol{p}_u) \tag{4}$$

Koren et al. [8], who won the Netflix Prize, introduced both SGD optimization and Alternating Least Squares (ALS) optimization, which alternately optimizes $\boldsymbol{q}_i$ and $\boldsymbol{p}_u$. They highlighted that matrix factorization can solve the curse of dimensionality caused by high-dimensional data, and achieved highly accurate recommendations, even for high-dimensional data.

In recent years, Hidasi et al. [9] proposed a recommender system employing users' data of all their past clicks on an e-commerce website with the Gated Recurrent Unit (GRU) model, which is a type of modern Recursive Neural Network (RNN). However, the techniques described above are not very effective for very sparse data. Therefore, all existing automatic recommender systems for real estate properties, such as Ietty, apply rule-based algorithms that reflect users' desired conditions.

2.2 Floor Plan Image Analysis

Research into property floor plans prior to the development of deep learning has been based on graphical analysis of floor plans. For example, Hanazato et al. [10] analyzed floor plans using adjacency graphs with the nodes labeled as "rooms" and "corridors". They used four divided datasets classified by the square area of the floor plans. They then examined patterns of adjacency graphs and the numbers of the patterns, and classified the patterns into six types according to the distance from each node to other nodes of the adjacency graphs. In addition, Takizawa et al. [11] analyzed the rental amounts of properties using adjacency graphs, again with labeled nodes, and with the edges labeled as "doors" and "windows". They employed floor plans of "3LDK," "3K," or "3DK" apartments in Kyoto, Japan, and extracted subgraphs from the adjacency graphs to effectively estimate the rent from the presence/absence of common subgraphs. However, these approaches involve extremely high costs to manually create adjacency graphs of floor plans.

†https://www.netflixprize.com/

Table 1 Layout type predicted by the method of [13]

layout type	#	layout type	#
2LDK	5194	1LDK	758
2DK	5083	2SLDK	507
3LDK	2985	1DK	388
1K	2419	4LDK	214
2K	2065	3K	140
3DK	1112	3SLDK	121
one room	974	others	180

Table 2 Room types in the floor plan predicted by [13]

name	explanation	name	explanation
Loft	**loft**	**Hall**	**corridor**
WR	**western room**	**PR**	**powder room**
Bal	**balcony**	CL	closet
UPDN	**stairs**	E	entrance
JR	**Japanese room**	DR	dress room
WIC	**walk-in closet**	L	living
Ver	**verandah**	D	dining
R	**room**	K	kitchen
BR	**bedroom**	DK	dining kitchen
UB	**modular bathroom**	LD	living-dining
Ba	**bathroom**	LDK	living-dining kitchen
WC	**toilet**	Other	others

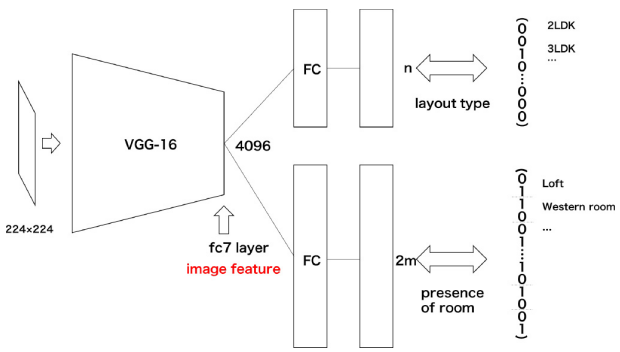


Fig. 1 Network architecture of FloorNet developed by [13]

Ohara et al. [12] developed a property search system that reflects users' preferences through common subgraphs of floor plans. Using the dataset created by [12], Takada et al. [13] estimated floor plan structures and retrieved similar floor plans with a query floor plan. Using the ImageNet [14] pre-trained model, they fine-tuned the model by multi-task learning to predict both layout type, provided in Table 1, and the presence of each room underlined in Table 2. The network architecture is shown in Fig. 1. Hereafter, we refer to their fine-tuned model as FloorNet. They solved the retrieval task using deep features extracted from the floor plan images by FloorNet. Specifically, they calculated the Euclidean distance between deep features of the query floor plan and those of each floor plan in the database. They then

Table 3 Classification of images associated with properties.

		predicted class									
		floor plan	living room	entrance	kitchen	bath room	rest room	wash room	view	equipment	other
actual class	floor plan	2997	0	0	0	0	0	0	0	0	3
	living room	1	2847	38	53	5	2	2	5	36	11
	entrance	0	107	2693	30	1	0	3	6	127	33
	kitchen	0	137	18	2814	0	0	1	1	11	18
	bathroom	0	5	1	2	2869	52	43	2	10	16
	restroom	0	3	3	0	44	2932	11	0	1	6
	washroom	0	16	10	5	97	20	2750	2	63	37
	view	0	54	3	2	1	0	0	2785	130	25
	equipment	0	137	127	37	16	4	39	205	2344	91
	other	14	691	188	127	38	53	113	54	914	808

obtained similar floor plans by the nearest neighbor search and evaluated them by the Maximum Common Subgraph (MCS) method [12]. FloorNet can extract deep features by considering floor plan structures without the need for any graphical annotation such as adjacency graphs extracted by hand.

3. Preliminary Experiments

In this study, we used floor plan images of real estate properties. However, images associated with properties often do not have labels such as "floor plan," "living room," and "kitchen"; therefore, we need to classify the images associated with the properties and extract only floor plan images. In this preliminary experiment, we used the Convolutional Neural Network (CNN); the trained classifier was used to reduce the property data to those containing the floor plans in the main experiments.

3.1 Dataset

In this preliminary experiment, we used property images from Ietty labeled into 10 different classes: "floor plan," "living room," "entrance," "kitchen," "bathroom," "restroom," "washroom," "view," "equipment," and "other." Note that they are different from the floor plan images used in the main experiments. We sampled 15,000 images for each class and split them into a training set and a test set at a ratio of 4:1. Hence, the training set contained 120,000 images and the test set contained 30,000 images.

3.2 Experiments

We classified the images associated with properties into the 10 classes using the ResNet-50-based network [15], whose final layer was changed to 10 dimensions. The model was pre-trained by ImageNet [14]. To improve generalization, we conducted data augmentation by horizontally flipping the images and cropping them randomly before performing 4-fold cross-validation.

Table 3 shows the results of the classification. The global accuracy was 86.1%. After checking all images whose actual label was "floor plan" but predicted as another label, and vice versa, the actual label turned out to be incorrect. Thus, we succeeded in fully identifying if an image was a "floor plan". As a result, we can conclude that we can accurately extract only floor plan images from the database and the whole pipeline proposed in this study can be performed automatically. Conversely, we achieved low accuracy for images labeled as "equipment" or "other" because of the within-class variance.

Table 4 Amount of each data type in the dataset used in this study.

	total	training set	validation set	test set
# evaluations	220,094	132,055	44,019	44,020
# users	19,538	11,425	4,357	6,703
# properties	131,947	79,769	32,284	27,847

4. Proposed Method

4.1 Dataset

Ietty, a real estate property rental company, holds attribute data of both real estate properties and users, and recommends several properties to users through a rule-based algorithm on its website. Furthermore, users can evaluate each recommended property by selecting either "want to see the property," " bookmark," or "no interest." In this study, we obtained evaluation data between 2016 and 2017 and reduced it to 220,094 cases that contain the properties with the floor plan images using the classifier trained by the preliminary experiment. Then, we split them into a training set, validation set, and test set at a ratio of 3:1:1. Thus, the dataset included attribute data of 19,538 users, and attribute data and floor plan images of 131,947 properties. More detail is shown in Table 4. The average number evaluations per property is less than 2, showing the sparsity of the evaluation data.

4.2 Overview of Proposed Method

We defined "want to see the property" and " bookmark" as positive user evaluations and "no interest" as negative user evaluations of the properties. We then designated these as users' preference for properties, which we predicted using the methods available for sparse data. In this study, we predicted preference by two methods: content-based filtering using similarities of both users and properties, and MLP using attribute data plus deep features extracted from floor

plan images as the input. As these methods can be used without previous user evaluation data, they are robust for sparse data. Namely, they are less sensitive to the cold-start problem. We also proposed a system called hybrid filtering that combines these two methods to predict users' preferences.

Although the layout type of floor plans, such as "two bedrooms + one bathroom," is included in the property attribute data, we suggest that the accuracy of preference prediction is improved by considering the actual floor plans. For example, within the same layout type, some users desire properties without a direct connection between the entrance and a child's bedroom so that young children cannot go outside without their parents' permission, while other users desire convenient properties that allow access to the corridor from each room. Therefore, we added the FloorNet features of floor plans to the MLP input to include a consideration of floor plan structure. The details of each method are described in the following subsections.

4.3 Content-Based Filtering

We predicted user preference by content-based filtering using the similarity of attributes for both users and properties. We defined $\boldsymbol{u}$ and $\boldsymbol{i}$ as the attribute data of a user and a property, respectively, and let the positive evaluation value be w and the negative evaluation value be -1. This w is a constant value expressing the ratio between the number of negative evaluations and positive evaluations in the training set. We employed each pair $(\boldsymbol{u}, \boldsymbol{i})$ in the training set whose evaluation value $v_{ui}(= w \text{ or } -1)$ is known and each pair $(\boldsymbol{u}_{test}, \boldsymbol{i}_{test})$ in the test set. We calculated the cosine similarity $\cos(\boldsymbol{u}_{test}, \boldsymbol{u})$ for each $\boldsymbol{u}$, and let U_{CBF} be the set of the top k_u% of all $\boldsymbol{u}$ by similarity. In the same way, we calculated the cosine similarity $\cos(\boldsymbol{i}_{test}, \boldsymbol{i})$ for each $\boldsymbol{i}$, and let I_{CBF} be the set of the top k_i% of all $\boldsymbol{i}$ by similarity. These k_u and k_i values are equivalent hyperparameters. Furthermore, we calculated the predicted evaluation value v_{CBF} for the pair $(\boldsymbol{u}_{test}, \boldsymbol{i}_{test})$ in the test set as (5).

$$v_{CBF} = \sum_{\boldsymbol{u} \in U_{CBF}} \sum_{\boldsymbol{i} \in I_{CBF}} \cos(\boldsymbol{u}_{test}, \boldsymbol{u}) \cdot \cos(\boldsymbol{i}_{test}, \boldsymbol{i}) \cdot v_{ui} \quad (5)$$

We then predicted the preference as positive if the predicted evaluation value v_{CBF} was larger than zero, and negative otherwise.

4.4 MLP

We also predicted user preference by MLP. To improve generalization, the MLP has both dropout [16] and batch normalization [17]. The network architecture of the MLP is shown in Fig. 2. As the input of the MLP, we used both the attribute data of users and properties and the deep features extracted from floor plan images of the properties. At the time of training, we used the training set and validation set whereas, and at the time of testing, we excluded the dropout and used the test set. From the final *fc* layer, a two-dimensional vector (x_n, x_p) corresponding to negative and positive evaluations was obtained as the output. We then predicted the preference as positive if the predicted evaluation value $v_{MLP} = x_p - x_n$ was larger than zero, and negative otherwise.

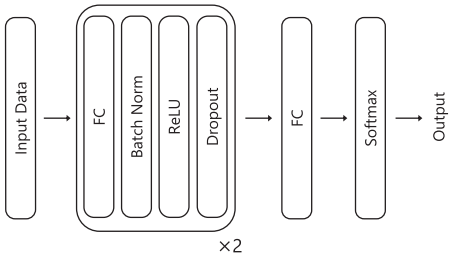


Fig. 2 Network architecture of the MLP.

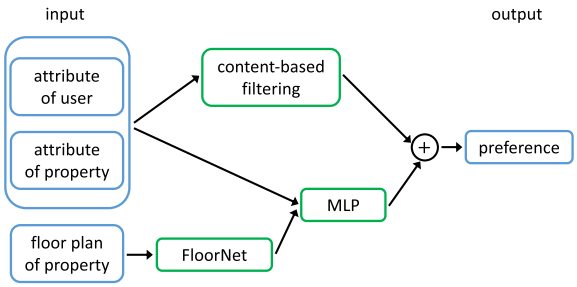


Fig. 3 Architecture of the hybrid filtering.

To extract the deep features of floor plan images, we used a model that was fine-tuned by the improved method of Takada et al. [13], FloorNet. Specifically, we used ResNet-50 [15] instead of VGG-16 [18]. Moreover, we used floor plan images randomly rotated for data augmentation as the input of FloorNet. The improved FloorNet was fine-tuned by multi-task learning in the same way as [13]. We used the fine-tuned model as the feature extractor and obtained the feature vectors of 2,048 dimensions from the *pool5* layer as the deep features of floor plans.

4.5 Hybrid Filtering

We then predicted user preference according to the weighted sum of two predicted evaluation values: v_{CBF} in Sect. 4.3 and v_{MLP} in Sect. 4.4. An overview of the architecture is shown in Fig. 3. We used the validation set instead of the test set in Sect. 4.3 and Sect. 4.4 and calculated the standard deviations σ_{CBF} and σ_{MLP} of predicted evaluation values for each method. Then, we defined their reciprocal numbers as w_{CBF} and w_{MLP}, respectively, and used them as the weight of each predicted evaluation value for the test set according to (6).

$$v_{HF} = w_{CBF} \cdot v_{CBF} + w_{MLP} \cdot v_{MLP} \quad (6)$$

By performing scaling, we obtained the predicted evaluation value v_{HF} that emphasizes the two methods to the same extent. Like other methods, we predicted the preference as

positive if the predicted evaluation value v_{HF} was larger than zero, and negative otherwise.

5. Experimental Results

5.1 Metrics

A total of 76,871 positive data (35%) and 143,223 negative data (65%) comprised the total 220,094 samples of evaluation data. Bias existed in the dataset; thus, we used the Matthews Correlation Coefficient (MCC) [2] as the evaluation metric to ensure equal evaluation. MCC is calculated as (7) where the number of true positives in the prediction result is TP, that of true negatives is TN, that of false positives is FP, and that of false negatives is FN. The maximum value of MCC is 1, the minimum value is -1, and a larger value indicates better performance.

$$\mathrm{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{7}$$

5.2 Content-Based Filtering

We predicted users' preference for properties using the content-based filtering method described in Sect. 4.3. We used the top k_u% (= 1, 10, 100%) of the similar users and the top k_i% (= 1, 10, 100%) of the similar properties in the training set. Figure 4 shows the result of the prediction.

The best performance is obtained when both k_u and k_i are 10%. If k_u is too small, it is difficult to accurately predict the preference because the number of users in the training set that can be referenced is too limited. Conversely, if k_u is too large, it is difficult to accurately predict the preference as a result of referencing users in the training set whose attributes are not completely similar to those of each user in the test set. Regarding k_i, if k_u is not large (k_u = 1, 10%), it is likewise preferable to choose a value that is not too small and not too large. However, if k_u is 100%, indicating that the information of all users in the training set is used, it is likely preferable to also use the information of all properties in the training set. The left of Table 5 shows the detailed results of the prediction by content-based filtering with k_u of 10% and k_i of 10%.

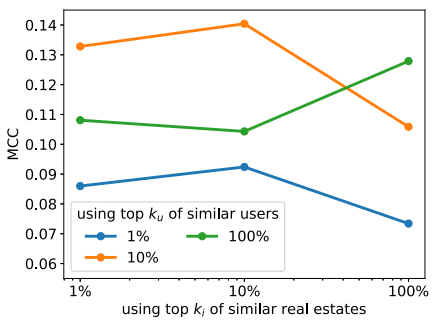


Fig. 4 Prediction results of users' preference for properties determined by content-based filtering.

5.3 MLP

We also predicted users' preference for properties using the MLP method described in Sect. 4.4 and compared the following three types of image feature:

attribute only
: Using only attribute data of users and properties

pre-trained
: Using attribute data of users and properties, and deep features extracted from floor plan images by the pre-trained model of ImageNet [14]

fine-tuned
: Using attribute data of users and properties, and deep features extracted from floor plan images by the FloorNet in Sect. 4.4

The dataset employed for the fine-tuning was the same as that used by Takada et al. [13]. It was created by Ohara et al. [12] and contains floor plan images from two sources: SUUMO[†] and HOME'S dataset[††]. The floor plan images are RGB image data rescaled to 224 × 224 pixels, preserving the aspect ratio by padding. Using the dataset, we fine-tuned FloorNet by multi-task learning in the same way as [13]. The tasks involved predicting both layout type provided in Table 1 and the presence of each room underlined in Table 2. The average validation accuracy for layout type and the presence of each room is 82.0% with the VGG-16-based network and 85.8% with the ResNet-50-based network. Therefore, we show the results using FloorNet based on ResNet-50 in this section. Further comparison between the VGG-16-based and ResNet-50-based networks is presented in Sect. 5.5.

We implemented both the network of FloorNet and that of the MLP using Chainer [19], [20]. We then fine-tuned FloorNet with a learning rate of 10^{-3} and trained the MLP with a learning rate of 10^{-2}. In both training experiments, the batch size was 32, the weight decay was 0.0005, and we adopted the SGD optimizer with a momentum of 0.9. The middle of Table 5 shows the results of the prediction and Fig. 5 shows the ROC curve of the content-based filtering, MLP without floor plan image features, and MLP including image features after fine-tuning.

Among the three types of input used in this study, the best performance of MLP is achieved by inputting deep features extracted from the floor plan images by the fine-tuned model. Therefore, floor plans seem to be effective for predicting users' preference for properties. In addition, the performance of MLP is higher using the fine-tuned model instead of the pre-trained model of ImageNet [14] as the extractor of deep features of floor plans. Hence, we suggest that better deep features can be extracted and floor plan

[†]http://suumo.jp/
[††]http://www.nii.ac.jp/dsc/idr/next/homes.html

Table 5 Prediction results of users' preference for properties determined by content-based filtering (*left*), MLP (*middle*), and hybrid filtering (*right*).

method	content-based	MLP			hybrid		
image feature	attribute only	attribute only	pre-trained	fine-tuned	attribute only	pre-trained	fine-tuned
MCC	0.140	0.127	0.142	0.149	0.150	0.159	**0.166**
accuracy	0.568	0.612	0.616	**0.619**	0.600	0.604	0.607
precision	0.417	0.439	0.446	**0.451**	0.437	0.442	0.446
recall	**0.593**	0.396	0.418	0.423	0.500	0.507	0.514
confusion matrix $\begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$	$\begin{pmatrix} 15870 & 12775 \\ 6254 & 9121 \end{pmatrix}$	$\begin{pmatrix} 20843 & 7802 \\ 9281 & 6094 \end{pmatrix}$	$\begin{pmatrix} 20678 & 7967 \\ 8954 & 6421 \end{pmatrix}$	$\begin{pmatrix} 20725 & 7920 \\ 8866 & 6509 \end{pmatrix}$	$\begin{pmatrix} 18748 & 9897 \\ 7693 & 7682 \end{pmatrix}$	$\begin{pmatrix} 18796 & 9849 \\ 7579 & 7796 \end{pmatrix}$	$\begin{pmatrix} 18823 & 9822 \\ 7479 & 7896 \end{pmatrix}$

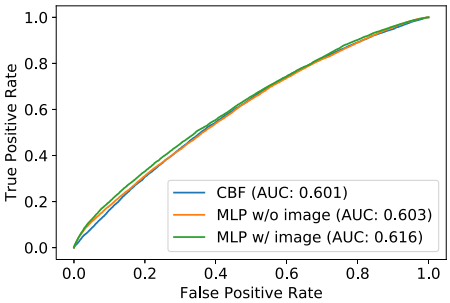


Fig. 5 ROC curve of content-based filtering and MLP results.

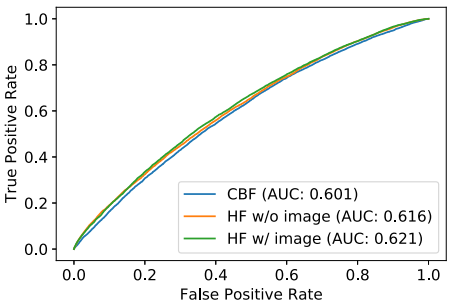


Fig. 6 ROC curve of content-based filtering and hybrid filtering.

features can be better expressed by fine-tuning. Moreover, Fig. 5 illustrates that the differences in the True Positive Rate (TPR, Recall) between MLP including image features after fine-tuning and the other methods are large when the False Positive Rate (FPR) is small (0.0–0.4). This indicates that including floor plans is especially effective for recommendations of a small number of properties.

5.4 Hybrid Filtering

Finally, we predicted users' preference for properties using the hybrid filtering described in Sect. 4.5 and compared the same three types of input as in Sect. 5.3. The MLP that formed part of the hybrid filtering was also the same as in Sect. 5.3. Regarding the content-based filtering that formed the other part of the hybrid filtering, we used the top $k_u = 10\%$ of similar users and the top $k_i = 10\%$ of similar properties in the training set. The right of Table 5 shows the results of the prediction and Fig. 6 shows the ROC curve of content-based filtering, hybrid filtering without floor plan image features, and hybrid filtering including image features after fine-tuning.

Table 6 Comparison results of the three methods employed in this study.

method	fine-tune	model	FAR↓	FRR↓	MCC↑
content-based	N	N/A	0.446	**0.407**	0.140
MLP	N	N/A	**0.272**	0.604	0.127
	Y	VGG-16	0.279	0.580	0.143
	Y	ResNet-50	0.276	0.577	0.149
hybrid	N	N/A	0.346	0.500	0.150
	Y	VGG-16	0.338	0.498	0.160
	Y	ResNet-50	0.343	0.486	**0.166**

Among the three types of input, the best performance of hybrid filtering, which is MCC of 0.166, is also achieved when we use deep features extracted from the floor plan images by the fine-tuned model. Thus, better deep features help predict user preference more accurately. Moreover, Fig. 6 illustrates that the difference of TPR between hybrid filtering with/without image features and content-based filtering is generally large. Therefore, hybrid filtering is effective in many situations.

5.5 Comparison

Table 6 compares hybrid filtering with content-based filtering and MLP, as well as the three types of fine-tuning (i.e., those not using images and CNNs, those using VGG-16-based networks, and those using ResNet-50-based networks). The best performance is achieved by hybrid filtering including deep features extracted from floor plan images by the fine-tuned model. When hybrid filtering is compared with MLP for each type of fine-tuning, it is clear that each MCC is considerably improved. On the other hand, the False Acceptance Rate (FAR) of MLP is better than that of hybrid filtering. Moreover, the False Rejection Rate (FRR) of content-based filtering is better than that of MLP and hybrid filtering. Therefore, MLP is relatively effective for finding users' favorite properties, and content-based filtering is relatively effective for finding properties in which users have no interest. Hybrid filtering has characteristics of both MLP and content-based filtering; thus, it combines the advantages of both methods to achieve better performance in terms of MCC.

Table 6 also compares the performances of ResNet-50-based FloorNet with those of VGG-16-based FloorNet. Typically, ResNet is known for extracting more robust features than VGG, and a slight improvement is observed for ResNet-50-based FloorNet in Table 6. Moreover, deep features extracted by ResNet-50-based FloorNet have 2,048

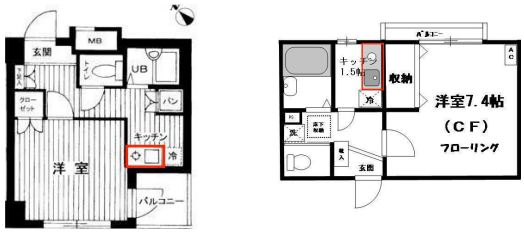


Fig. 7 Example floor plan images of properties whose user evaluations were correctly predicted regardless of whether floor plan images were used (GT: negative (*left*), positive (*right*)). Stoves are surrounded by a red rectangle.

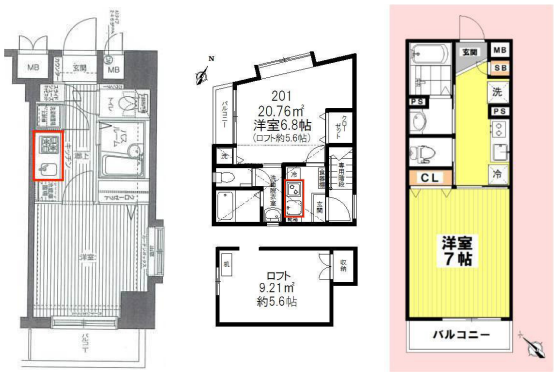


Fig. 8 Example floor plan images of properties whose user evaluations were predicted differently depending on whether floor plan images were used (GT: negative (*left*), positive (*middle*), negative (*right*)). *Left* and *middle* represent improved prediction and *right* represents worse prediction.

dimensions while those of VGG-16-based FloorNet have 4,096 dimensions; therefore, the number of MLP parameters is reduced. Specifically, the MLP using ResNet-50-based FloorNet has 3.3 M parameters while that using VGG-16-based FloorNet has 5.3 M parameters; thus, an almost 40% reduction of parameters is achieved.

In order to support our claim that using images and FloorNet is effective for predicting users' preference for properties, we present concrete examples of floor plan images of properties whose user evaluations were predicted by hybrid filtering. Figure 7 shows floor plan images of properties whose user evaluations are correctly predicted, regardless of whether floor plan images are used. The user evaluated the left image as negative and the right as positive because the user wanted the kitchen to be a separate room with a door. Figure 8 shows floor plan images of properties whose user evaluations are predicted differently depending on whether floor plan images are used. The left and middle images are better examples of using floor plan images. The user evaluated the left image as negative; using the floor plan image with FloorNet results in correct prediction whereas not using the image results in incorrect prediction. Furthermore, the user evaluated the middle image as positive; using the floor plan image results in correct prediction. One of the reasons for improved prediction seems to be that using floor plan images reveals if the kitchen is a separate room. There are a total of 11 examples where using the floor plan images improves the prediction for this user. The image on the right is the only example of worse prediction when using a floor plan image. The user evaluated the right image as negative, yet an incorrect prediction was generated when the floor plan image was used. A possible reason for this is that the pink background around the floor plan confuses FloorNet. Such a special case is difficult to deal with, despite the fact that the color of floor plan images typically helps FloorNet recognize the structures of floor plans.

6. Conclusions

In this study, with the aim of developing a recommender system for special items that are not mass-produced, we predict and compare users' preference for real estate properties using three methods: content-based filtering, MLP, and hybrid filtering. Content-based filtering employs the similarities of both users and properties. MLP employs attribute data of users and properties as well as deep features extracted from floor plan images using CNN as the input. Hybrid filtering is a combination of the two methods. The best performance was achieved by hybrid filtering that included FloorNet features extracted from floor plan images by the fine-tuned model.

There are two limitations to this study. First, we used the method proposed by Takada et al. [13] to extract the deep features of floor plans; there are also other approaches for this purpose. For example, Liu et al. [21] succeeded in converting a rasterized floor plan image into a vector graphics representation. This method is composed of multiple stages, and their discriminative network, which is used to extract junction layers from input floor plan images, may be able to be applied to our proposed method. Second, for actual property recommendations, the speed of processing is important. The content-based filtering used in this study is likely to be computationally intensive. Therefore, it should be made more efficient for application to actual property recommendations.

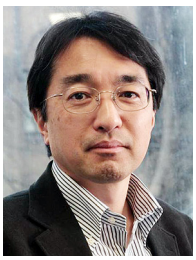
References

[1] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Commun. ACM, vol.35, no.12, pp.61–70, Dec. 1992.
[2] B.W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," Biochimica et Biophysica Acta (BBA) - Protein Structure, vol.405, no.2, pp.442–451, 1975.
[3] N. Kato, T. Yamasaki, K. Aizawa, and T. Ohama, "Users' preference prediction of real estates featuring floor plan analysis using floornet," Proc. 2018 ACM Workshop on Multimedia for Real Estate Tech, RETech'18, New York, NY, USA, pp.7–11, ACM, 2018.
[4] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI), pp.43–52, Morgan Kaufmann Publishers Inc., 1998.
[5] D.M. Pennock, "Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach," Proc. Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI), pp.473–480, Morgan Kaufmann Publishers Inc., 2000.
[6] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z.

Chen, "Scalable collaborative filtering using cluster-based smoothing," Proc. 28th Annual ACM SIGIR Conf. on Research and Development in Information Retrieval, 2005.
[7] S. Func, "Netflix update: Try this at home," 2006. http://sifter.org/simon/journal/20061211.html.
[8] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol.42, no.8, pp.30–37, 2009.
[9] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," Proc. International Conference on Learning Representations (ICLR), 2015.
[10] T. Hanazato, Y. Hirano, and M. Sasaki, "Syantic analysis of large size condominium units supplied in the tokyo metroporitan area," Journal of Structural and Construction Engineering, vol.70, no.591, pp.9–16, 2005.
[11] A. Takizawa, K. Yoshida, and N. Kato, "Applying graph mining to rent analysis considering room layouts," Journal of Environmental Engineering, vol.73, no.623, pp.139–146, 2008.
[12] K. Ohara, T. Yamasaki, and K. Aizawa, "An intuitive system for searching apartments using floor plans and areas of rooms," The 78th national convention of IPSJ, 5Y-08, 2016. (in Japanese).
[13] Y. Takada, N. Inoue, T. Yamasaki, and K. Aizawa, "Similar floor plan retrieval featuring multi-task learning of layout type classification and room presence prediction," IEEE International Conference on Consumer Electronics (ICCE), pp.931–936, 2018.
[14] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, pp.1097–1105, 2012.
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, IEEE Computer Society, 2016.
[16] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," Journal of machine learning research, vol.15, no.1, pp.1929–1958, 2014.
[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International Conference on Machine Learning (ICML), pp.448–456, 2015.
[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Proc. International Conference on Learning Representations (ICLR), 2015.
[19] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS), 2015.
[20] Y. Niitani, T. Ogawa, S. Saito, and M. Saito, "Chainercv: a library for deep learning in computer vision," ACM Multimedia, pp.1217–1220, 2017.
[21] C. Liu, J. Wu, P. Kohli, and Y. Furukawa, "Raster-to-vector: Revisiting floorplan transformation," The IEEE International Conference on Computer Vision (ICCV), pp.2214–2222, Oct. 2017.

Naoki Kato received the B.E. degree in information and communication engineering from the University of Tokyo. He is currently a master's student of the University of Tokyo. His recent research interest is in zero-shot learning and few-shot learning in computer vision. He is a student member of IEEE and ITE.

Toshihiko Yamasaki received the B.S. degree in electronic engineering, the M.S. degree in information and communication engineering, and the Ph.D. degree from The University of Tokyo. From April 2004 to Oct. 2006, he was an Assistant Professor at Department of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo. He is currently an Associate Professor at Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo. He was a JSPS Fellow for Research Abroad and a visiting scientist at Cornell University from Feb. 2011 to Feb. 2013. His current research interests include attractiveness computing based on multimedia big data analysis, computer vision, pattern recognition, machine learning, and so on. Dr. Yamasaki is a member of IEEE, ACM, AAAI, IEICE, ITE, IPSJ, and JSAI.



Kiyoharu Aizawa received the B.E., the M.E., and the Dr.Eng. degrees in Electrical Engineering all from the University of Tokyo, in 1983, 1985, 1988, respectively. He is currently a Professor at Department of Information and Communication Engineering of the University of Tokyo. He was a Visiting Assistant Professor at University of Illinois from 1990 to 1992. His research interest is in multimedia applications, image processing and computer vision. He received the 1987 Young Engineer Award and the 1990, 1998 Best Paper Awards, the 1991 Achievement Award, 1999 Electronics Society Award from IEICE Japan, and the 1998 Fujio Frontier Award, the 2002 and 2009 Best Paper Award, and 2013 Achievement award from ITE Japan. He received the IBM Japan Science Prize in 2002. He is on Editorial Boards of IEEE MultiMedia, ACM TOMM, APSIPA Transactions on Signal and Information Processing, and International Journal of Multimedia Information Retrieval. He served as the Editor in Chief of Journal of ITE Japan, an Associate Editor of IEEE Trans. Image Processing, IEEE Trans. CSVT and IEEE Trans. Multimedia. He is/was a president of ITE and ISS society of IEICE, 2019 and 2018, respectively. He has served a number of international and domestic conferences; he was a General co-Chair of ACM Multimedia 2012 and ACM ICMR2018. He is a Fellow of IEEE, IEICE, ITE and a council member of Science Council of Japan.

Takemi Ohama received the M.A. degree in Education from Tokyo Gakugei University. He belonged in Nikkei Research Inc., Yahoo Japan Corporation, Groupon Japan, Inc., and Marketing Applications, Inc. He is currently Chief Technical Officer and General Manager of Engineering Division at Ietty Co., Ltd.