Social Experiment on Advisory Recommender System for Energy-Saving

Hiroki Shigeyoshi[1], Ken'iti Tamano[1], Ryosuke Saga[1], Hiroshi Tsuji[1], Shuki Inoue[2], and Tsuyoshi Ueno[3]

[1] Osaka Prefecture University, 1-1, Gakuen-cho, Naka-ku, Sakai-shi, Osaka, Japan
shigeyoshi@mis.cs.osakafu-u.ac.jp, tama@a7m3.jp, {saga,tsuji}@cs.osakafu-u.ac.jp

[2] Energy Use R&D Center, The Kansai Electric Power Co., Inc., 3-11-20 Nakoji, Amagasaki-shi, Hyogo, Japan
Inoue.shuuki@d4.kepco.co.jp

[3] Central Research Institute of Electric Power Industry, 2-11-1, Iwadokita, Komae-shi, Tokyo, Japan
ueno@criepi.denken.or.jp

Abstract. This paper describes a social experiment on an advisory recommender system for home energy-saving, called KNOTES. Based on the user's value sense and the effectiveness of the advice, KNOTES aims to recommend highly effective advices over the user's own preferences. In addition, KNOTES uses an advice reference history to avoid the repetition of redundant advice. For the social experiment, forty-seven subjects used KNOTES for about two months. Introducing four metrics for comparing KNOTES with a random recommender, this paper verifies that KNOTES could recommend the advices which are desirable from the view of energy-saving and could avoid the repetition of redundant advices. The remaining issue has been prediction of the users' preferences according to their value sense.

Keywords: recommender system, home-energy-saving, man-machine interaction, knowledge management.

1 Introduction

Various recommender systems have been proposed to help users effectively select contents that interest them from a potentially overwhelming set of choices [1]. The traditional systems have usually dealt with business products such as books and music, focusing on users' preferences that show what they are most likely to accept. Meanwhile, they are not designed for a recommendation of advices that change and improve our lifestyle, such as energy saving, weight control, and smoking stoppage. In such cases, effective recommendations are not always based on the user's preferences as the best results cannot be achieved when they choose only the advice that they like. Not all preferred advice is effective for all users. Thus, it is desirable that the systems focus on not only user's preference but also the effectiveness of advice.

S. Yamamoto (Ed.): HIMI/HCII 2013, Part I, LNCS 8016, pp. 545–554, 2013.

For evaluating such advisory recommender systems, it is necessary to collect log data by social experiments. Based on the advice execution history from the log data, the effect of advice can be verified. It is also necessary to simultaneously define the evaluation metrics by considering the system characteristics [2].

This paper describes an advisory recommender systems and its social experiment. The system is named KNOTES (KNOwledge & Transaction based domestic Energy saving support System), which we developed in the previous study [3]. This system specializes in energy saving and aims to select effective advices that are in user's interest. The experiment includes forty seven subjects who used KNOTES for about two months. Analyzing the log data, KNOTES was evaluated with the proposed evaluation metrics.

In chapter 2, this paper will introduce the traditional recommender systems and requirements for advisory recommender systems. In chapter 3, this paper will show overview of an advisory recommender system for energy-saving and its recommendation algorithm. In chapter 4, this paper will describe the social experiment, which verifies whether the purposes of the system are accomplished. In chapter 5, this paper will conclude and suggest the future issues.

2 Recommender Systems

2.1 Overview of Related Systems

By offering useful information, recommender systems support users in various decision-making processes, such as what books to buy, what music to listen or what online news to read [4]. There are two major traditional recommendation methods. One is content-based filtering, which chooses content that is similar to the user's current interests. The other is collaborative filtering, which chooses content based on the interests of similar users. Both methods attempt to predict users' interest in an item by focusing on their preferences.

Accuracy metrics are widely used as an evaluation tool for systems based on such methods by expressing how precisely the system can predict the user's preference. These are arguably the most important metrics, because there is marginal use of recommendations for content that does not interest the user [5].

In recent attempts to improve the user's satisfaction for the systems, other metrics have attracted attention [2], [6], focusing on novelty, serendipity [7], and diversity [6]. However, when these systems tried to simultaneously improve several metrics, a trade-off problem occurred [4]. In particular, accuracy decreased as the system put a higher priority on novelty, thus, it is essential to improve several metrics while maintaining accuracy.

In addition, it was also pointed out that the performance of the system changes according to the number of users and the number of items [8].Therefore, it is necessary to select recommendation methods that are most appropriate to the purpose of the system and data size [2].

2.2 Requirements for Advisory Recommender Systems

In this section, we consider the case in where a recommender system provides advice which changes and improves our life-style, such as energy-saving, weight control and smoking stoppage. It was reported that energy saving, such as time restricting the use of an air-conditioner, reduces comfort [9]. Such advice is often disliked by a user because of the mental workload. If the system makes recommendations only according to the user's preferences, some effective advices might not be suggested, thus, it is necessary to consider not only user's preference but also the effectiveness of advice. The advisory recommender system should suggest advice which is acceptable for a user and yet desirable for energy-saving.

It is suggested that users dislike getting the same advice. This problem prevents the users from repeatedly using the system. Thus, it is desirable to avoid repeating advice which a user has already followed and advice and overly repetitive advice.

The most important requirement is that the system prompts users to act on the advice given, because the essence of recommender systems is to support users in decision-making processes. Implementing this requirement imposes a change of consciousness to the user, thus, it also seems to be the most difficult issue. Encapsulating the above discussion, there are three requirements on advisory recommender systems:

- To recommend highly effective advice in the user's interest,
- To avoid the repetition of redundant advice,
- To prompt user to execute the given advice.

3 Advisory Recommender System on Energy-Saving

3.1 System Structure

To implement these requirements, we developed an advisory recommender system named KNOTES [3]. This system deals with energy-saving advice. An overview of the system is shown in Figure 1. First, users input their profile data including value sense and appliances owned by them. Next, they are required to input their monthly energy consumption about electricity, gas, and kerosene. Based on the data, KNOTES gives advice to users and records its recommendation logs. If the user follows the advice given in recommendations, the execution logs record it. The user can simultaneously evaluate the advice on a scale of 1–5 on how easy it felt to comply and how likely he would be to recommend it to others. Users can repeatedly receive different recommendations.

To reflect the user's preferences in recommendations, KNOTES collects data about user value sense using questionnaires, allowing the system to predict which advice will be favorable. The value sense and the effectiveness in advice data are used to recommend highly effective advice that remains in the user's interest. An advice reference history, including recommendation logs and execution logs, is used to avoid the repetition of redundant advice.

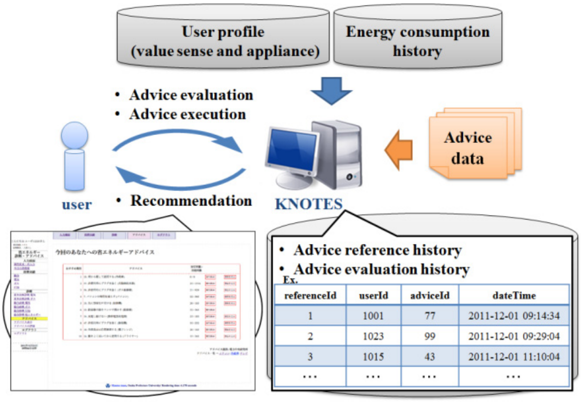

Fig. 1. Advisory recommendation from KNOTES

3.2 Advice Features

Advice in KNOTES has attributes such as those shown in Table 1. The advice data is based on data that was introduced in the report [10]. "*Energy-saving*", "*CO_2 reduction*", and "*Cost saving*" show the prospective amounts saved when the advice is executed once. "*Maximum number of available times*" shows the number of times that the advice is ideally executed in one year, thereby avoiding the repetition of advice to avoid user resistance. "*Difficulty level*" and "*Recommended-level*" are dynamic scores of 1–5 that are updated to be a mean value by advice evaluations. "*Easiness*" is the combination of "*Maximum number of available times*" and "*Difficulty level*". "*Energy type*" shows the energy target of the advice: electricity, gas, and kerosene. If the energy target is not recognized, "*no data*" is inputted. "*Season*" shows when the advice should be executed.

Table 1. Example of advice data

Suggestion	Appliance	Energy saving [MJ/time]	CO_2 reduction [g/time]	Cost saving [yen/time]	Easiness: Maximum number of available times [times/year]	Easiness: Difficulty level (easy:1, difficult:5)	Recommended level (unrecommend:1, recommend:5)	Energy type	Season
Use a saving water shower	Water heater	2,549.00	130,772.00	7,257.00	1	5.0	3.0	No data	Whole year
Reduce the cooling time	Electric fan	0.22	10.53	0.53	112	3.0	3.0	Electricity	Summer
Reduce the heating time	Electric stove	3.47	162.08	8.08	169	3.0	3.0	Electricity	Winter
…	…	…	…	…	…	…	…	…	…

3.3 Algorithm for Recommendation

KNOTES calculates the scores to select advice from all advices user by user. According the score, advices in top *n* (*n* is 10 in social experiment of chapter 4) are selected to become recommendations. The algorithm is divided into the following five steps.

I. Based on user value sense and the effectiveness of advice, the system scores each advice.

First, based on user value sense given in user profile, KNOTES calculates the weight v_i of each attribute i (i is in the range 1–5): "*Energy saving*", "*CO_2 reduction*", "*Cost saving*", "*Easiness*", and "*Recommended-level*". These weights express how important the user considers the five attributes. Then, the system calculates the score p_j of advice j (j is in the range of $1 - N$ (N is a 104 in the social experiment of chapter 4)) with the weight v_i and the attribute data a_{ij} using formula (1). The total of the weights is normalized as one. The maximum data in each attribute is normalized as one for calculation.

$$p_j^1 = \sum_i a_{ij} v_i \tag{1}$$

In this calculation, "*Easiness*" has two attributes, "*Maximum number of available times*" and "*Difficulty level*", which are reversal values. Thus, they are subtracted from each maximum data before normalization. The normalized "*Difficulty level*" is multiplied by the normalized "*Maximum number of available times*", and the value is used as "*Easiness*". In this step, a score of the advice that matches user value sense and have high effects will be raised.

II. Considering user energy consumption volume, each score is revised.

Based on the ratio of the energy consumption in the last month c_1 and that in the same month of the previous year c_0, each score is revised by the following formula (2).

$$p_j^2 = (c_1/c_0)p_j^1 \tag{2}$$

III. Recognizing user's own appliances and season, the system chooses available advices.

The advice about unowned appliances and off-season advice is not available.

$$\begin{cases} \text{owned appliance} & p_j^3 = p_j^2 \\ \text{not owned appliance} & p_j^3 = 0 \end{cases} \tag{3}$$

$$\begin{cases} \text{on season} & p_j^4 = p_j^3 \\ \text{off season} & p_j^4 = 0 \end{cases} \tag{4}$$

IV. Referring to advice reference histories, each score is revised.

Using formula (5), advice seldom executed has a greater chance of being recommended. In addition, the system can avoid the repetition of the same advice over the available time. For advice j, m_j is the maximum number of available time, and x_j is the number of executed times.

$$p_j^5 = (m_j - x_j)p_j^4/m_j \tag{5}$$

The system can avoid repeating the same advice. k is a decay constant from 0 to 1 (k is 0.95 in the social experiment of chapter 4). d_j is the number of recommended times on advice j. With this calculation, the score p_j decreases as d_j increases.

$$p_j^6 = k^{d_j} p_j^5 \tag{6}$$

V. At random, a score of the advice is raised.

To promote energy saving, it is desirable to inform users of every available advice. Thus, the system tries to recommend even low effect advice at least once a year. With the following formula (7), the system chooses one advice and raises its score at random with the 25% probability.

$$p_j^7 = p_j^6 + \{0, 1, 2, 3, 4, 5\} \tag{7}$$

4 Social Experiment

4.1 Overview of Experiment

A social experiment was conducted to verify the three requirements for advisory recommender systems in chapter 2. This section describes the data collection and the evaluation metrics according to the requirements.

Data Collection. First, we distributed the manual and questionnaires to forty seven subjects to collect the users' data. The questionnaires included questions about appliances owned by users, value sense, among others. They also included questions about the monthly amounts of energy consumption and bills for electricity, gas, kerosene from September 2009 to August 2011. This energy consumption data was used for recommendation algorithm in formula (2). Every subject answered these questionnaires by October 21, 2011.

Next, each subject used KNOTES online from December 1, 2011 to February 8, 2012, and its action logs were simultaneously recorded in the system. In this experiment, 10 suggestions were recommended from a total of 104 pieces of advice at once. Moreover, each subject was prescribed to input the monthly amounts of energy consumption into the system. Thus, it was expected that every subject would use the system once a week during experiment.

To collect user ratings for each advice, we performed our investigation from February 1, 2012 to February 8, 2012. Each subject answered a question about user rating (want to execute: 5, not: 1) on a web site. The rating data was transformed into a binary scale by converting every rating of 4 or 5 to "*like*", and those of 1–3 to "*dislike*".

Evaluation Metrics. To verify the three requirements, this subsection proposes four evaluation metrics: accuracy, excess, achievement and accumulation recall. Then, a random recommender system is introduced for a comparative evaluation with KNOTES.

- To recommend highly effective advice in the user's interest

It is necessary to identify effective advices. Thus, all advice is divided into two equal groups, "*high effect*" and "*low effect*", according to the sum of attributes in the advice data: "*Energy saving*", "*CO_2 reduction*" and "*Cost saving*". The proportion of "*like*" and "*high effect*" advices in all of the recommended advices is calculated to verify this requirement.

Accuracy is defined as confirmation that the system can reflect the user's preferences. The mean absolute error and mean square error have been used widely as accuracy metrics [11]. However, the metrics have been useful only when the system predicts user rating of each suggestion. Therefore, the metrics are not useful for KNOTES. Here accuracy is defined as a mean of user ratings in recommended advices.

$$\text{Accuracy} = \text{Mean of user ratings in recommended advices} \tag{8}$$

- To avoid the repetition of redundant advice

Excess is defined as the sum of ratios of the number of excess times to the maximum number of available times on each advice, as shown in formula (9). If the user executes the advice more than the ideal times, the user may not efficiently save energy. Excess becomes better as the value gets closer to zero. For advice j for user u, E_u is the set of available advice, m_j is a maximum number of available times and x_j^u is a number of executed times.

$$\text{Excess} = 100\frac{1}{|E_u|}\sum_{j\in E_u}\frac{max\left(m_j, x_j^u\right) - m_j}{m_j} \tag{9}$$

- To prompt user to execute the given advice

For the verification of this requirement, achievement and accumulation recall are defined. Achievement is defined as the sum of ratios of the number of executed times to the maximum number of available times of each advice. For the user, it is desirable to execute all available advices according to the maximum number of available time on each advice.

$$\text{Achievement} = 100\frac{1}{|E_u|}\sum_{j\in E_u}\frac{min\ (m_j, x_j^u)}{m_j} \tag{10}$$

Accumulation recall is defined as the ratio of the sum of the number of executed times to the sum of the maximum number of available times on each "*like*" advice. T_u is the set of "*like*" available advice for user u.

$$\text{Accumulation recall} = 100\frac{\sum_{j\in T_u} min\ (m_j, x_j^u)}{\sum_{j\in T_u} m_j} \tag{11}$$

To comparatively evaluate KNOTES, a random recommender system is used. Recognizing user's own appliances and season, the random recommender also chooses available advices, as shown in formulas (3) and (4). Then, the system selects 10 suggestions at random from the available advices.

For three metrics, excess, achievement and accumulation recall, it is necessary to calculate the number of executed times x_j^u. This number is calculated using the advice execution ratio in the social experiment.

4.2 Results

Experimental Results

As a result of the social experiment, twenty seven subjects were regarded as valid data. The average subject used the system six times during experiments, as shown in Table 2. This table shows the top and bottom three users in descending order by the number of times the system was used. The execution ratio varied user by user and showed marginal correlation with the number of times the system was used.

Table 2. Results of social experiment on KNOTES (top and bottom three users)

User id	Number of use times	Types of available advice	Types of like-advice	Total number of recommended advice (Types)	Total number of executed advice (Types)	Execution ratio
1003	16	65	12	160 (45)	11 (9)	6.9%
1001	15	74	17	150 (44)	53 (16)	35.3%
1024	13	79	30	130 (44)	77 (18)	59.2%
…	…	…	…	…	…	…
1048	2	79	37	20 (14)	12 (8)	60.0%
1014	1	63	25	10 (10)	0 (0)	0.0%
1034	1	74	20	10 (10)	0 (0)	0.0%
mean	6.3	74.4	30.4	62.6 (23.1)	18.3 (7.7)	26.5%

Evaluation Results. The proportions of "*like*" and "*high effect*" advice in all recommended advice are shown in Table 3, along with the values of the t-test in each proportion. The "*like*" and "*high effect*" advices were likely to be recommended in KNOTES more than when using the random recommender. Effective advice was given high priority compared with interesting advices in KNOTES, because the proportions of "*high effect*" and "*dislike*" advice were better than those of "*low effect*" and "*like*" advices.

The results for four metrics are shown in Table 4. From t-value, the differences between KNOTES and the random recommender were not proven for the following metrics: accuracy, achievement and accumulation recall. The excess metrics was better than the random recommender, and the difference between the systems was also proven. The value of excess in KNOTES was zero, therefore, it was verified that the system could avoid the repetition of redundant advice. However, the accuracy of KNOTES was not better than that of the random recommender.

Table 3. Proportions in recommended advice

	high effect		low effect	
	like	dislike	like	dislike
KNOTES	31.87%	43.64%	6.95%	17.54%
Random	20.93%	23.57%	20.02%	35.49%
t-value	4.61	11.07	−8.69	−11.01

Table 4. Results for four metrics

	Accuracy	Excess [%]	Achievement [%]	Accumulation recall [%]
KNOTES	2.41	0	2.54	0.11
Random	2.51	0.85	2.36	0.07
t-value	-0.99	-2.84	0.89	1.86

4.3 Discussion

In results, the number of times KNOTES was used was fewer than expected, and the execution ratio was not good, possibly because the interface of KNOTES was difficult to use. Moreover, users hesitated to execute an advice because of the mental workload required. In an advisory recommender system, it is important that the system prompts users to execute advice in a user-friendly way.

Conversely, it is conceivable that the advisory recommender system should store several small advices for recommendation. Some energy-saving methods have not only a significant effect but correspondingly significant drawbacks. By collecting many small advices, it will become easier to reflect the life rhythm and demand of the user in making recommendation. It will also become easier to recommend advices at more opportune times for the user.

KNOTES was likely to recommend "*like*" and "*high effect*" advices more than the random recommender, as shown in Table 3. This is because the system focuses on not only user's preference but also the effectiveness of advice in first step of recommendation algorithm. However, at the same step, the system failed to predict user's preferences by using user value sense. It resulted in a decline in accuracy, as shown in Table 4.

To precisely predict user's preferences, it is desirable to investigate the tendency of the user from log data. Analyzing what advice is more readily accepted by a user is regarded as a future issue. This social experiment provided useful data for such an analysis. Moreover, it is considered as a remedy to combine the traditional method focusing user's preferences with KNOTES.

5 Conclusions

An advisory recommender system that provides advice in the domain of energy saving, weight control and smoking stoppage is required. Unlike the traditional recommender systems, the system needs to focus on not only user's preference but also the effectiveness of advice. Moreover, the system should be user-friendly by the avoidance of the repetition of redundant advice. To implement the requirements, this paper has described an interactive system, named KNOTES, and its social experiments. The social experiment was conducted with forty seven subjects for about two months. To verify the requirements, the evaluation metrics have been defined: accuracy, excess, achievement and accumulation recall.

It has verified that KNOTES recommended the "*high effect*" advices and avoided the repetition of redundant advice. Meanwhile, it has not verified that the system recommend "*like*" advices, because of inaccuracy in the prediction of users' preferences based on their value sense. Improving the accuracy of the recommendations is one of remaining issues by applying the traditional method to KNOTES.

References

1. Resnick, P., Varian, H.R.: Recommender Systems. Communications of the ACM 40(3), 56–58 (1997)
2. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating Collaborative Filtering Recommender System. ACM Transactions on Information Systems 22(1), 5–53 (2004)
3. Shigeyoshi, H., Inoue, S., Tamano, K., Aoki, S., Tsuji, H., Ueno, T.: Knowledge and Transaction Based Domestic Energy Saving Support System. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part IV. LNCS, vol. 6884, pp. 242–251. Springer, Heidelberg (2011)
4. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommender Systems Handbook (2010)
5. Swearingen, K., Sinha, R.: Beyond Algorithms: An HCI Perspective on Recommender Systems. In: ACM SIGIR Workshop on Recommender Systems, New Orleans, Louisiana, vol. 13, pp. 393–408 (2001)
6. Ziegler, C., McNee, S.M., Konstan, J.A., Lausen, G.: Improving Recommendation Lists Through Topic Diversification. In: 14th International Conference on World Wide Web, Chiba, Japan, pp. 22–32 (2005)
7. Sawar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: 10th International Conference on World Wide Web, Hong Kong, China, pp. 285–295 (2001)
8. Ben Schafer, J., Konstan, J.A., Riedl, J.: E-commerce Recommendation Applications. Data Mining and Knowledge Discovery 5, 115–153 (2001)
9. Fanger, P.O.: Thermal Comfort. Danish Technical Press, Copenhagen (1970)
10. Ueno, T., Nakano, Y.: An Approach for Ranking Energy-Saving Activities in Residential Buildings -Characteristics of the Model-, CRIEPI Report: R06006, Japan (2007); Breese, J., Herlocker, J., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: 14th Conference on Uncertainty in Artificial Intelligence, Madison, Wisconsin, USA, pp.43–52 (1998)