A Ranking Method based on Users' Contexts for Information Recommendation

Kenta Oku[1], Shinsuke Nakajima[1], Jun Miyazaki[1],
Shunsuke Uemura[2], and Hirokazu Kato[1],

[1] Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma City,
Nara, 630-0101, Japan

{kenta-o, shin, miyazaki, kato}@is.naist.jp

[2] Nara Sangyo University,
12-1, Tatsuno-kita 3-chome, Sango-cho, Ikoma-gun,
Nara, 636-8503, Japan

uemurashunsuke@nara-su.ac.jp

ABSTRACT

We propose a ranking method using a Support Vector Machine for information recommendation. By using the SVM, a recommendation method can determine suitable items for a user from enormous item sets. However, it can decide based on just two classes: whether the user likes a thing or not. When there is a large number of recommended items, it is not easy for the user to find the best item by herself. To resolve this issue, it is desirable to rank the items based on the user's preferences. Moreover, the user's preferences change depending on the context. Based on the above problem, we propose a context-aware ranking method for information recommendation. Our method considers a user's context when ranking items. Our method consists of the following two steps: (1) Predicting important feature parameters for the user. (2) Calculating a ranking score of each item in recommendation candidates. In this paper, we describe our method and show experimental results.

Keywords

ranking, SVM, context-awareness, information recommendation.

1. INTRODUCTION

Information recommendation is a popular topic in the information explosion era. A recommendation method provides users with suitable items from enormous item sets. One of the ways to achieve such a method is to construct a user's preference model with machine learning algorithms. By using the user's model, the recommendation method can recommend suitable items for the user's preferences. We proposed recommendation methods applying a Support Vector Machine (SVM) [1], which is one of the major classification methods for two-class problems to construct a user's preference model [2] [3]. We also showed that it is possible to recommend suitable items for not only a user's preferences but also the user's contexts by considering the user's contexts when constructing the user's preference models.

Although suitable items for a user are determined by using the SVM, it can decide based on just two classes: whether the user likes something or not. When there is a large number of recommended items, it is not easy for the user to find the best item by herself. To solve this problem, it is desirable to rank the candidates in the order of how the user likes the items and show the top *n* items to the user.

In the field of document retrieval, some ranking methods using an SVM have been proposed [4] [5] [6]. By using the SVM, a decision plane is constructed in feature vector space to divide instances into two classes. Hirao et al. [4] use the Euclidean distance from the decision plane to each instance to rank the instances. However, there is no theoretical basis for the distance being used to rank instances. Weston et al. [5] proposed a multi-class SVM. It solves multi-class problems with an SVM. The multi-class SVM can be applied as a ranking method by regarding classes as ranks. Herbrich et al. [6] proposed a ranking SVM. The method is a kind of pair-wise training approach. Each instance pair is used for training. The method trains the relations between instances with the SVM.

Both the multi-class SVM and the ranking SVM are based on the assumption that there are relative order relations between instances. In recommendation problems, however, the above assumption is not assured. Since a user's preferences are changed depending on the user's contexts [2] [3], it is not always that there are relative relations between items evaluated in different contexts. Thus, these methods are not suitable for such recommendation problems.

In this paper, based on the above problem, we propose a context-aware ranking method for information recommendation. The method considers a user's contexts when ranking items. Our method consists of the following two steps:

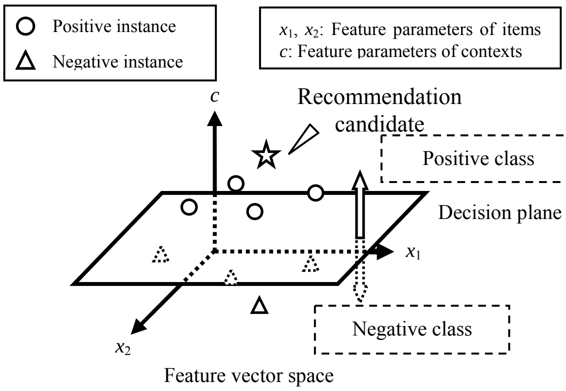


Figure 1. Concept diagram of user's preference model using an SVM.

1. Predicting important feature parameters for the user.
2. Calculating a ranking score of each item in recommendation candidates.

A lot of feature parameters are used to construct a user's preference model. In the first step, our method predicts important feature parameters for the user's current contexts from the user's model. In the second step, based on the user's important parameters, the method calculates a ranking score of each item to rank recommendation candidates.

The rest of this paper is organized as follows. In Section 2, we describe a context-aware recommendation method using an SVM we proposed, and some ranking methods using the SVM. In Section 3, we describe a context-aware ranking method for information recommendation. Experimental results are shown in Section 4. Finally, Section 5 concludes the paper.

2. RELATED WORK

2.1 Context-aware recommendation method using SVM

We proposed a context-aware recommendation method using an SVM [2] [3]. The SVM [1] is one of the major classification methods for two-class problems. We applied it to construction of a user's preference model used for recommendation. Figure 1 shows a concept diagram of this model.

The model consists of multi-dimensional feature vector space. There are some training instances to construct the model. Each instance is represented by p-dimensional feature vectors of items and q-dimensional feature vectors of a user's contexts. Each instance is labeled with "positive" or "negative," indicating "like" or "dislike." Then a decision plane is constructed to divide the instances into a positive class or negative class.

Figure 1 shows an example of the model that consists of two-dimensional feature vectors of items and a one-dimensional feature vector of the user's contexts. The symbols "○" and "Δ" denote positive instances and negative instances respectively. A decision plane is constructed between the positive and the negative instances. If new items are plotted in the positive side at a user's current contexts on this model, the items are regarded as recommendation candidates for the user. By using such a method, it can recommend suitable items not only for a user's preferences but the user's contexts as well.

However, it can decide by only two classes; whether a user likes something or not. When there is large number of recommended items, it is not easy to find the best item by herself. To solve this issue, it is desirable to rank the candidates in the order of how the user likes the items and show the top n items to the user.

2.2 Ranking method using SVM

In the field of document retrieval, they proposed some ranking methods using an SVM [4] [5] [6].

Hirao et al. [4] proposed an SVM-based important sentence extraction technique. They use $g(\boldsymbol{x})$ the Euclidean distance from a decision plane to $\boldsymbol{x}$ to rank the sentences. However, there is no theoretical basis on which $g(\boldsymbol{x})$ can be used to rank instances.

Weston et al. [5] proposed a multi-class SVM. It solves multi-class problems with an SVM. The approach to solving k-class problems is to consider the problems as a collection of binary classification problems. In this method, k classifiers are constructed. For example, the n^{th} classifier constructs a decision plane between class n and $k - 1$ other classes. The multi-class SVM can be applied as a ranking method by regarding classes as ranks.

Herbrich et al. [6] proposed a ranking SVM. The method is a kind of pair-wise training approach. Each instance pair is used for training. The method trains the relations between instances with the SVM.

Both the multi-class SVM and the ranking SVM are based on the assumption that there are relative order relations between instances. In recommendation problems, however, the above assumption is not assured. Since user's preferences are changed depending on the user's contexts [2] [3], it is not always that there are relative relations between items evaluated in different contexts. Thus, these methods are not suitable for such recommendation problems.

In this paper, based on the above problem, we propose a context-aware ranking method for information recommendation.

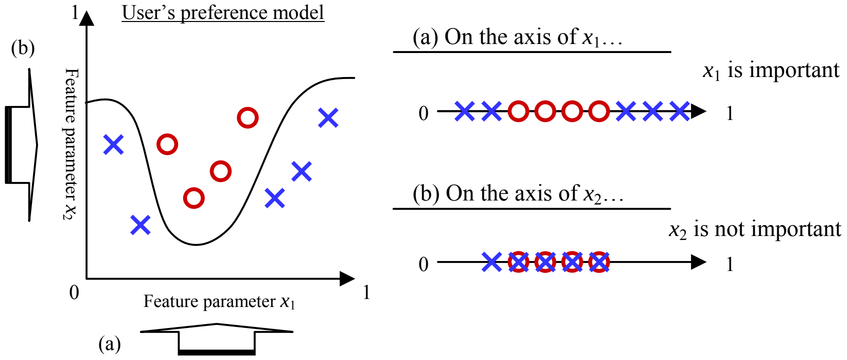


Figure 2. A basic idea to predict important feature parameters from a user's model.

3. CONTEXT-AWARE RANKING METHOD

We propose a context-aware ranking method for information recommendation. Recommendation candidates are decided by using a context-aware recommendation method described in Section 2.1. Our ranking method ranks the recommendation candidates in the order of how a user likes the items. This method consists of the following two steps:

1. Predicting important feature parameters for the user.
2. Calculating a ranking score of each item in recommendation candidates.

A lot of feature parameters are used to construct a user's preference model. In the first step, our method predicts important feature parameters for the user's current contexts from the user's model. In the second step, based on the user's important parameters, the method calculates a ranking score of each item to rank recommendation candidates.

3.1 Predicting important feature parameters for the user

A user will evaluate items based on the user's sense of values. If a recommendation method can understand which factors are important for the user, the method can rank items based on the factors. For example, if a user regards "cost" of restaurants as important, the method ranks restaurants based on their cost.

In this section, we propose a method predicting important feature parameters for the user from the user's model.

3.1.1 Basic idea

We show an example of a user's preference model consisting of two-dimensional feature vector space in Figure 2. Training instances are represented by two-dimensional feature parameters. The symbols "○" and "×" denote positive and negative instances. A decision plane is constructed between them.

Now, we select just one feature parameter x_1. On the axis of x_1, the positive instances and the negative instances are distributed as shown in Figure 2 (a). On the other hand, we also select feature parameter x_2. On the axis of x_2, the instances are distributed as shown in Figure 2 (b). The positive and the negative instances are separated in Figure 2 (a), while they are intermingled in Figure 2 (b). What we can see from this example is that it is possible to decide the class of the instances by the value of x_1, while it is hard to decide them by the value of x_2. Based on these views, a method analyzes how the i^{th} feature parameter x_i contributes to decision of positive instances or negative instances. Finally, it is possible to predict which feature parameters are important for the user.

3.1.2 Contribution rate of feature parameter

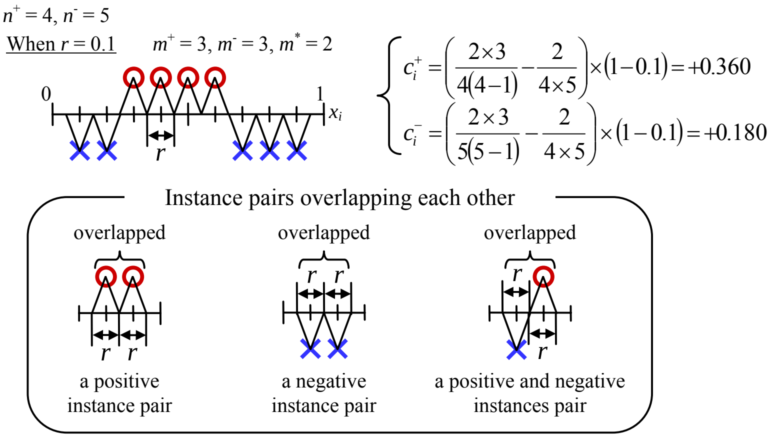
To find important feature parameters based on the above basic idea, we define a contribution rate as a measure indicating how the feature parameters contribute to decision of the positive or negative instances. Let c_i^+ (c_i^-) be a contribution rate which denotes how the i^{th} feature parameter x_i contributes to decision of the positive (negative) instances. We formulate c_i^+ and c_i^- as follows:

$$c_i^+ = \left(\frac{2m^+}{n^+\left(n^+ - 1\right)} - \frac{m^*}{n^+ n^-}\right)(1-r) \tag{1}$$

$$c_i^- = \left(\frac{2m^-}{n^-\left(n^- - 1\right)} - \frac{m^*}{n^+ n^-}\right)(1-r) \tag{2}$$

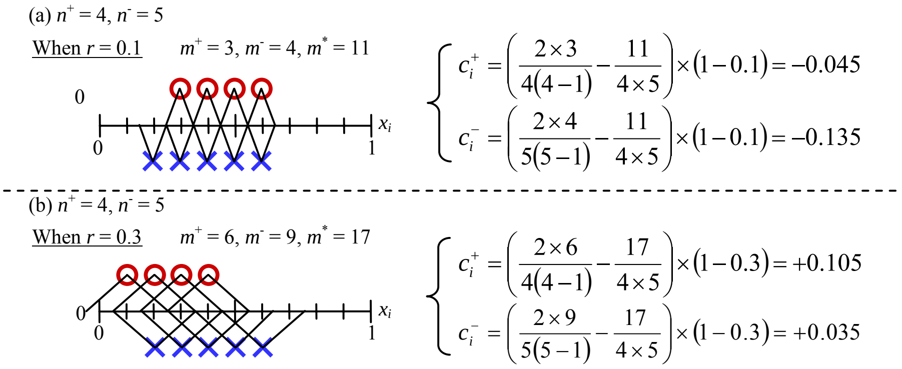
Figure 3 shows an example of a contribution rate when i^{th} feature parameter x_i is selected. Here, n^+ and n^- denote the number of positive and negative instances. As shown in Figure 3, when considering r $\{r \mid 0 \leqq r \leqq 1\}$ a proper range of an instance, we regard any two positive (negative) instances overlapping each other as a positive (negative) instance pair. And we regard one positive instance and one negative instance overlapping each other as a positive and negative instance pair. Then m^+ and m^- denote the number of positive instance pairs and negative instance pairs. And m^* denotes the number of positive and negative instance pairs. In the example of Figure 3, $n^+ = 4$, $n^- = 5$. And $m^+ = 3$, $m^- = 3$ and $m^* = 2$ when $r = 0.1$. Then $c_i^+ = 0.360$ and $c_i^- = 0.180$.

The first term in Equation (1) (Equation (2)) denotes the percentage of the number of positive (negative) instance pairs in all numbers of their pairs. Like Figure 2 (a), if the positive (negative) instances are more gathered, the value of this term is higher. The second term denotes the percentage of the number of positive and negative instances pairs in all numbers of their pairs. Like Figure 2 (b), if the positive and the negative instances are more intermingled, the value of this term is higher. Consequently, c_i^+ (c_i^-) denotes the difference between the values of the first and



$$\begin{cases} c_i^+ = \left(\frac{2\times 3}{4(4-1)} - \frac{2}{4\times 5}\right)\times(1-0.1) = +0.360 \\ c_i^- = \left(\frac{2\times 3}{5(5-1)} - \frac{2}{4\times 5}\right)\times(1-0.1) = +0.180 \end{cases}$$

Figure 3. An example of a contribution rate to positive and negative instances and instance pairs overlapping each other.



$$\begin{cases} c_i^+ = \left(\frac{2\times 3}{4(4-1)} - \frac{11}{4\times 5}\right)\times(1-0.1) = -0.045 \\ c_i^- = \left(\frac{2\times 4}{5(5-1)} - \frac{11}{4\times 5}\right)\times(1-0.1) = -0.135 \end{cases}$$

$$\begin{cases} c_i^+ = \left(\frac{2\times 6}{4(4-1)} - \frac{17}{4\times 5}\right)\times(1-0.3) = +0.105 \\ c_i^- = \left(\frac{2\times 9}{5(5-1)} - \frac{17}{4\times 5}\right)\times(1-0.3) = +0.035 \end{cases}$$

Figure 4. Examples of calculation of the contribution rates to the positive and the negative instances.

the second term. We interpret that when $c_i^+ > 0$ and $c_i^- > 0$, the feature parameter x_i contributes to decision of the positive or negative instances. Let r^* be the minimum r on the condition that $c_i^+ > 0$ and $c_i^- > 0$ and then let c_i^+ and c_i^- with r^* be the positive and negative contribution rates of x_i. In particular, when $c_i^+ = 1$ and $c_i^- = 1$, it is possible to decide the positive or negative instances by just the value of x_i while when $c_i^+ \leqq 0$ or $c_i^- \leqq 0$, it is hard to decide them by the value of x_i.

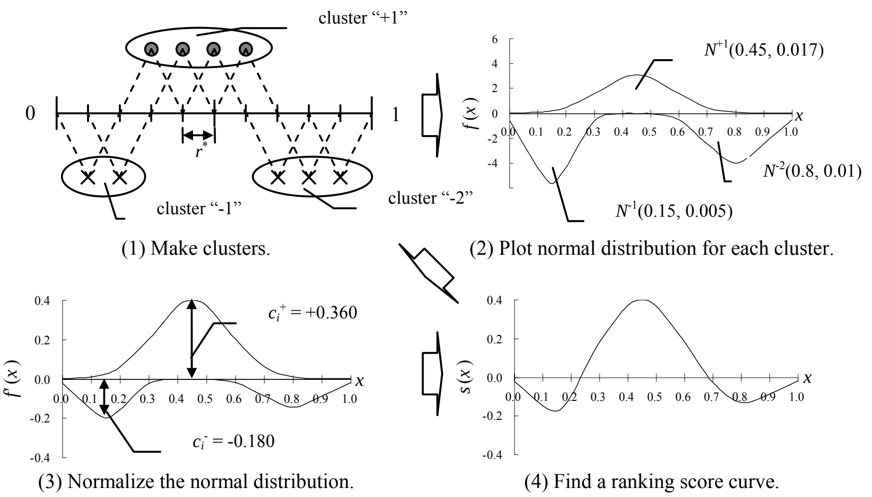
For reference, we show the other examples of calculation of the contribution rates in Figure 4. In the case of Figure 4 (a), $c_i^+ \leqq 0$ and $c_i^- \leqq 0$, since positive and negative instances are intermingled. Accordingly, it is hard to decide the positive or negative instances by the value of x_i. In the case of Figure 4 (b), although $c_i^+ > 0$ and $c_i^- > 0$, their values are lower than the case of Figure 3. This means that the x_i does not contribute to decision of the positive or negative instances as much as the case of Figure 3.

3.1.3 Contribution rate of feature parameter

The contribution rates are calculated based on training instances on the user's model. However, the issue is that it depends on the user's contexts which feature parameters are important for the user. The target training instances must be selected considering the issue. One of the ways to select training instances is based on the similarity between the user's current contexts and the feature vector of contexts in training instances.

3.2 Calculating a ranking score of each item in recommendation candidates

In Section 3.1, the important feature parameters are predicted by analyzing how the positive (negative) instances are gathered. In this section, the method calculates a ranking score depending on the value of x_i by analyzing where the positive (negative) instances are gathered. Figure 5 shows the example of the process to calculate a ranking score. The procedure consists of the following steps:



(1) Make clusters.

(2) Plot normal distribution for each cluster.

(3) Normalize the normal distribution.

(4) Find a ranking score curve.

Figure 5. Examples of calculation of a ranking score.

1. Regard that the positive (negative) instance pairs overlapping each other when considering r^* belong to the same cluster. In the case of Figure 5, three clusters ("+1", "-1" and "-2") are made.
2. Suppose that those instances in each cluster are distributed following normal distribution $N(\mu, \sigma)$ and plot the normal distribution for each cluster. In the case of Figure 5, three normal distributions ($N^{+1}(0.45, 0.017)$, $N^{-1}(0.15, 0.005)$ and $N^{-2}(0.8, 0.01)$) are plotted. Here, the normal distribution of the cluster by positive instances is plotted on the positive side, and those by negative instances are plotted on the negative side.
3. Normalize the distribution on the positive (negative) side by replacing the maximum (minimum) value with c_i^+ (c_i^-).
4. Find one curve by compounding both the distribution on the positive and the negative side.

We define $s(x_i)$ the ranking score curve as the curve found in step 4. The curve is based on normal distributions as shown in Figure 5. The reason why we use the normal distribution is based on the following assumption: When a value of x_i is given as a peak point ("0.45" the mean of the instances in the cluster "+1"), a user may be satisfied with the items the best. But the user's satisfaction may be attenuated as the value is apart from the peak point. Based on the assumption, in this paper, we briefly use the normal distribution $N^{+1}(0.45, 0.017)$ for the cluster "+1". In the same way, we use the normal distributions $N^{-1}(0.15, 0.005)$ and $N^{-2}(0.8, 0.01)$ for the clusters "-1" and "-2". Although we use the normal distributions for such a problem, we would like to examine whether it is appropriate to the problem in future works.

The value of $s(x_i)$ depends on the value of x_i. In the case of Figure 5, examples are as follows:

$$x_i = 0.45 \rightarrow s(x_i) = s(0.45) = +0.360\left(= c_i^+\right) \quad (3)$$

$$x_i = 0.15 \rightarrow s(x_i) = s(0.15) = -0.180\left(= -c_i^-\right) \quad (4)$$

Finally, our ranking method calculates S_k the sum of $s(x_i)$ for each feature parameter.

$$S_k = \sum_{i=1}^{p} s(x_i) \quad (5)$$

The S_k means a ranking score of item k in recommendation candidates. The ranking method ranks items in recommendation candidates in the order of the S_k, and then shows the top n items to a user.

4. EXPERIMENTAL RESULT

In this section, we show experimental results applying our method as a restaurant recommendation method. We evaluate the effectiveness of our proposed method by comparing with the following methods.

I. Context-aware ranking method (our proposed method described in Section 3).

II. Multi-class SVM [5] (described in Section 2.2).

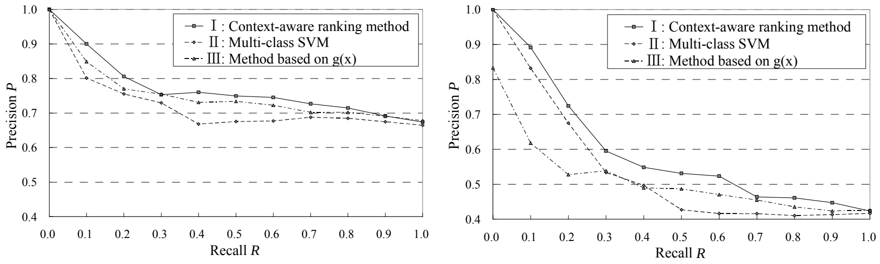
III. Method based on $g(\boldsymbol{x})$ the Euclidean distance from a decision plane to $\boldsymbol{x}$ in feature vector space of the SVM [4] (described in Section 2.2).

4.1 Procedure of Experiment

In this experiment, we applied our method to a restaurant recommendation system. We use the restaurant data set provided by "Yahoo! Gourmet in Japan [7]." Each restaurant data is represented by 28-dimensional feature parameters and a user's context is represented by 24-dimensional feature parameters.

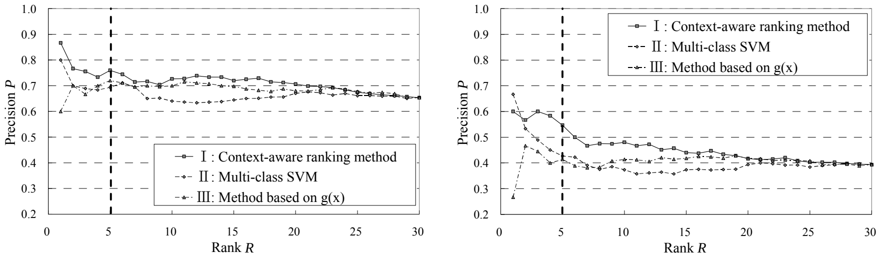
The number of users participating in this experiment is nine. The procedure to make their training data set is as follows:

1. Each user makes five context patterns.
2. 20 restaurant data are shown at each context pattern to the user.



(a) Regard three-grade instances as positive ones. (b) Regard four-grade instances as positive ones.

Figure 6. ***Recall-precision*** **curve.**



(a) Regard three-grade instances as positive ones. (b) Regard four-grade instances as positive ones.

Figure 7. ***R-precision curve.***

3. The user evaluates each restaurant data by giving five grades (dislike 1-2-3-4-5 like).

Each training instance is represented by 28-dimensional feature parameters of restaurant data and 24-dimensional feature parameters of the user's contexts. Each instance is labeled with five grades according to the level of the user's tastes. Consequently, the training data set consists of 100 instances. In the same way, the evaluation data set consists of 150 instances (30 restaurant data per a context pattern).

For each feature parameter of the restaurant data, a ranking score curve is calculated based on the user's training data set. Then a ranking score is calculated for each restaurant data in the evaluation data set.

We use *recall-precision* curve and *R-precision* for evaluation. When the top R^{th} items are recommended to the user, *recall* and *precision* are formulated as follows:

$$recall = \frac{\text{the number of positive instances in recommendations}}{\text{the total number of positive instances in evaluation data set}} \quad (\mathbf{6})$$

$$precision = \frac{\text{the number of positive instances in recommenda tions}}{\text{the total number of instances in recommenda tions}} \quad (\mathbf{7})$$

R-precision is the *precision* when the top R^{th} items are recommended. *Recall-precision* curve is a graph that shows the relation between *recall* and *precision*.

4.2 Results and Consideration

Figures 6 and 7 show *Recall-precision* curve and *R-precision* of each ranking method. The results are shown in the following cases:

- (a) Regarding instances evaluated over three-grade out of five as positive ones.
- (b) Regarding instances evaluated over four-grade out of five as positive ones.

These results show the average of results obtained for five context patterns.

In all cases, it is shown that our method is effective comparatively.

For *R-precision*, we can see from Figure 7 that a user may be satisfied with 75% out of five recommended items in the case of the above (a), and more satisfied with 55% of them in the case of the above (b). This result shows that our method works well enough for recommendation.

In the case of using a method based on $g(\boldsymbol{x})$ of the SVM, the method cannot distinguish between scores 3, 4 and 5 since they are regarded as positive instances evenly. Although a multi-class SVM is better than using the method based on $g(\boldsymbol{x})$, it is not better than our method.

In contrast, our proposed method shows effectiveness. It can rank the recommendation candidates considering the user's contexts

5. CONCLUSION AND FUTURE WORK

This paper proposes a context-aware ranking method for information recommendation and shows its effectiveness from experimental results. Although a user's preferences are changed depending on the user's contexts, the proposed method can rank recommendation candidates considering a user's contexts and preferences appropriately.

To consider the user's contexts, we select training instances just based on the similarity with feature parameters of the user's current contexts for calculation of ranking scores. However, it is not always that a user regards all the feature parameters of the user's contexts as important. Some parameters are important, others are not important for the user's choice of items. In future work, we would like to discuss selection of training instances considering feature parameters of a user's important contexts.

6. ACKNOWLEDGMENTS

This work was partly supported by MEXT (Grant-in-Aid for Scientific Research on Priority Areas #19024058), and by METI (Information Grand Voyage Project: "Profile Passport Service").

7. REFERENCES

[1] Cortes, C. and Vapnik, V.: Support-Vector Networks, Machine Learning, Vol. 20, No. 3, pp. 273-297 (1995).

[2] Oku, K., Nakajima, S., Miyazaki, J. and Uemura, S.: Investigation for Designing of Context-Aware Recommendation System Using SVM, The 2007 IAENG International Conference on Internet Computing and Web Services (IMECS 2007), pp. 970-975 (2007).

[3] Oku, K., Nakajima, S., Miyazaki, J. and Uemura, S: Context-Aware SVM for Context-Dependent Information Recommendation, Proc. of International Workshop on Future Mobile and Ubiquitous Information Technologies (FMUIT 2006) (2006).

[4] Hirao, T., Isozaki, H., Maeda, E. and Matsumoto, Y.: Extracting Important Sentences with Support Vector Machines, Proc. 19th Int. Conf. on Computational Linguistics, pp. 342-348 (2002).

[5] Weston, J. and Watkins, C.: Multi-class support vector machines, Technical report csd-tr-98-04, Royal Holloway, University of London, Surrey, England, (1998).

[6] Herbrich, R., Graepel, T. and Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. Advances in Large Margin Classifiers, pp. 115-132 (2000).

[7] Yahoo Japan, Yahoo! Gourmet (in Japan), http://gourmet.yahoo.co.jp/gourmet/.