Recommendation System on Mobile Devices using Estimated Geographical Search Conditions based on User Operations

Kaori KOBAYASHI† Daisuke KITAYAMA†† Kazutoshi SUMIYA††
Naiwala P. CHANDORASIRI††† Kazunari NAWA†††

† Graduate School of Human Science and Environment, University of Hyogo,
1-1-12 Shinzaike-Honcho, Himeji, Hyogo, 670-0092, Japan
†††Toyota Info Technology Center, co., LTD 6-6-20 Akasaka Minatoku Tokyo 107-0052, Japan

E-mail: † nd10r013@stshse.u-hyogo.ac.jp, ††{dkitayama, sumiya}@shse.u-hyogo.ac.jp
†††{np-chandrasiri, nawa}@jp.toyota-itc.com

Abstract We propose a geographical information recommender system for use on mobile devices based on user search operations in local services such as Yelp. We aim for intelligent interaction such as optimizing the search properties to estimate user requirements from realizable user operations on a general interface. If a user searches for restaurants in an unfamiliar location under a new category, it may be difficult to find a suitable restaurant on a small mobile screen. If the user region and category criteria can be estimated, we consider that the properties change according to the operation and number of viewed objects. We presume that we can estimate the search conditions using user operations in a particular region and category by analyzing user operations.

Keyword GIS, online maps, mobile phones, local search.

1. Introduction

There are various types of local services on the Web for obtaining geographical information on hotels, restaurants, and other locations. These services can also be used from mobile devices. For example, there are services that use maps and categories, such as Yahoo! Local, Google Places, and Yelp. In these services, to find a suitable target object, users have to specify multiple search properties such as the required region, category, price, score, and situation. Items such as region, category, price, and score are called "properties." Restaurants are termed "objects." When users specify these properties clearly, they can obtain adequate results. In existing systems, if too many objects are found in a particular region, a conventional system adjusts the number of displayed objects based only on the distance of the center point on the map and the object ratings. Contrariwise, when there are few or no displayed objects, a conventional system does not adjust the search properties.

We determined that a conventional system has two particular problems:

- If a user searches for a restaurant in an unknown location under categories they are unfamiliar with, the user may have difficulty specifying the properties clearly. When the user searches for a target restaurant, they blindly specify a certain region and categories.
- When there are too many displayed objects and the system adjusts the number of objects using object ratings, the user may not always want only highly rated objects. When there are few displayed objects, the user has to move to a new region and change the selected categories.

If a user searches for a restaurant in an unfamiliar location under unknown categories, it may be difficult for the user to find a suitable restaurant on a small mobile screen. If the user criteria of the region and category can be estimated, we may consider that the properties change according to the operation and number of viewed objects. We presume that we can estimate the search conditions using user operations for a region and category by analyzing the user operations. Therefore, we focus on the user operations in regions and categories for geographical information in local search services. Generally, there are four operations, changing the region, selecting the categories, viewing the objects, and selecting an object. We use the user operations to extract important factors attached to the user. Hence, we propose a geographical information recommender system for use in mobile devices based on user search operations. Moreover, we aim for intelligent interaction such as optimizing the search properties to estimate the user requirements from realizable user operations on a general interface.

The remainder of this paper is organized as follows: We describe the overview and other work related to this topic in section 2. Next, in section 3, we explain how we analyze the user operations. In section 4, we describe our algorithm for estimating the search conditions. In section 5, we describe the developed prototype system. Finally, we evaluate and discuss our experimentation in section 6.

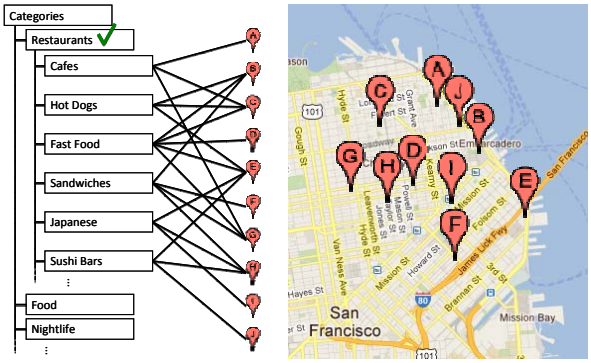


Figure 1: Category structure and an online map

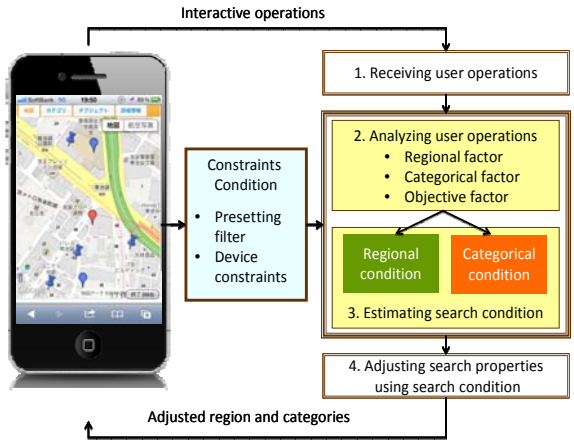


Figure 2: Conceptual image of recommendation system based on user operations

2. Our Approach

2.1. Overview of mobile recommender system with geographical search condition

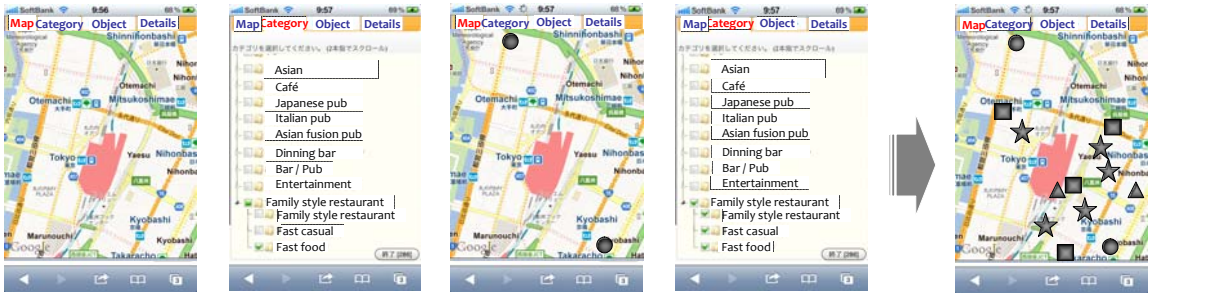
We consider it necessary to optimize the search criteria based on the state of factors of the local services. There are two types of search criteria: strong and weak. Strong criteria disallow variations from a specified value. Weak criteria use a given value as an example.

First, we describe the details of the user operation of the proposed mobile recommender system. The user operates three types of content: online maps, category hierarchy, and detailed information of the geographical objects. The user can operate such content to search for target objects. In an online map, changing the region implies that the user performs either a pan, centering, zoom in, or zoom out of a region on an online map (see Figure 1). In the category hierarchy, selecting categories implies that the user taps the target categories from a category list. In the detailed information of geographical objects, viewing objects implies that the user taps the objects displayed as icons on an online map and views the details of the object information in an object list. Selecting objects implies that the user saves candidate objects by tapping a save button for the selected objects in the object list. Objects can belong to multiple categories. For example, McDonalds belongs to the categories “Fast food,” “Sandwich,” and “Cafe” (see icon G in Figure 1).

In this way, users can explicitly set the search properties for the regions and categories. In our study, we adjust the user requirements using the search property, search condition, and constrained condition. The search property of a region is the search area, such as “San Francisco.” On the other hand, the search property of a category is a search category such as “Fast Food.” Here, the user can search for target restaurants using their selected categories and region. A search condition consists of the degree of strictness for the criteria of a region and categories. There are two kinds of search criteria, strong and weak. Constrained conditions have a preset filter such as an upper price limit, or device constraints such as the number of displayed objects depending on the screen size. Therefore, a constrained condition can always be satisfied by adjusting the search properties.

Next, we describe the process used in our system. First, our system extracts the user operations. Second, the system analyzes the user operations and extracts the operation factors. Third, we estimate the search conditions from the user operations. Finally, the system determines the display control based on the search conditions, and then recommends geographical information. Figure 2 shows the system architecture.

Next, we describe the operation factors used for estimating the search condition. 1. Estimate the degree of criterion strictness of a region from the changing region. This is called the regional factor, and is in either an ON or OFF state. 2. Estimate the degree of criterion strictness of categories from the object selection. This is called the categorical factor, and is also in either an ON or OFF state. 3. Estimate the degree of criterion strictness of both the region and categories from the object selection. This is called the objective factor, and SELECT, VIEW, and NONE are the available options. In this paper, we set the number of displayed objects to within a certain range as constrained conditions because users may find it hard to



1) Selecting the "Fast food" category

2) Viewing the map

3) Adding the other categories ("Casual dining")

4) Viewing the map

5) Automatically adding the categories similar to selected category

Figure 3: An example of recommendation by our system

obtain relevant objects when there are too many or too few displayed objects. Therefore, we developed an adding and reducing function for the number of displayed objects based on the search condition of the regions and categories. For example, if the state of a region has strong criteria based on the user preferences, we add/reduce objects using categories. In this way, we can recommend suitable geographical objects for an unfamiliar area and categories.

2.2. Motivating Example

Next, we provide some typical examples (see Figure 3). When a user wants to search for restaurants for a light meal near a certain station, the user will select a category such as "light meal." However, if there are few restaurants belonging to this category in the display region, the user can add other categories similar to "light meal" because the displayed results were not satisfactory. If the displayed results do not change significantly, however, it may be difficult for the user to search for target restaurants for a light meal if the user does not know other similar categories. In this case, the user cannot search for restaurants well, which is adapted into the user's search condition because the conventional system supports no relevant options. Therefore, our system approximates the user requirements based on the constrained condition of displayed numbers by estimating the user's search condition. In this case, the system adds a restaurant belonging to categories similar to "fast food" automatically. Additionally, the system adds restaurants to offer an appropriate number of displayed objects.

In another instance, a user may set the region and use "curry" as the category when the user wants to eat curry in an unfamiliar area. The user wants to know which regions have curry restaurants. Thus, the user selects various regions, as they are unfamiliar with the area. In this case, a conventional system does not support this even if the user looks all over for curry restaurants by changing the regions. Consequently, our system scales down the region to where multiple restaurants belonging to the curry category are located. In addition, the system scales down the number of restaurants appropriately. In this way, the system supports a user who is unfamiliar with setting the search conditions.

2.3. Related Work

Toda et al. [6] proposed a ranking method considering the proximity between the name of a location and the keywords associated with the keywords of the query. In this paper, we focus on recommending geographical information using the dynamic intensity of the user criteria during local search operations. Most of this research is related to information retrieval using user preferences. Teevan et al. [5] proposed an algorithm for personalizing retrieval results using user profiles based on both searching and creating documents. Martinez et al. [3] proposed a restaurant recommender system that hybridizes the collaborative knowledge-based system, REJA, which utilizes user preferences and creates profiles. These studies are similar to our attempt to relate geographical information with a user's static information. However, based on the operation of the user, we recommend object optimized by estimating the user's search criteria. Several studies related to information retrieval through manipulating online maps to extract the user's intention have been conducted. Weakliam et al. [7] proposed a system that generates personalized maps based on a user's map manipulations and defined map operations. These studies are similar to ours in the sense of extracting a user's intention from their map usage patterns. However, our system extracts the intensity of a user's search criteria from the map operations and category selection history.

Aula et al. [1] analyzed the user's behaviors when Web-search retrieval does not perform well. They conduct an additional query and natural language behaviors when a user cannot search for information effectively. In a local

search, adding categories and scaling down the regions are considered as behaviors. In conventional studies, the number of displayed objects from local services on mobile devices is controlled by an object score and the distance from the center position of the map [2, 4]. However, users have to specify the search criteria to manually control the number of displayed objects. Nowadays, conventional services can only display objects by setting the search criteria as per the user's requirement. However, important factors in the retrieved information may change after the user's operations on the displayed regions and the objects belonging to the selected categories.

3. Analyzing User Operations

3.1. Extraction of Regional Factors

We describe how we detect the user criteria of a region. We consider the user has certain criteria when searching in a certain region. We define the regional factors, as these criteria can be detected by the map operations.

When a user operates a map several times and the regions selected frequently overlap, we consider that the user wants to search in this overlapping region (see Figure. 4). At this time, we detect regional factors as the user's regional criteria. On the other hand, when a user operates a map several times, and there are few overlapped regions, we consider that this operation is for moving to a new area. In this case, the regional factor is not considered a regional criterion.

Algorithm 1 Extraction of regional factor

```
Require: n = 10, α = 3
  loop
    Waiting for regional input
    Regional factor is OFF
    R ← Current displayed region
    c ← 0
    for i = 0 to Length(Rs) do
      if R overlaps Rs[i] then
        c ← c + 1
      end if
    end for
    if c > α then
      Regional factor is ON
    end if
    R is added Rs
    if Length(Rs) > n then
      Rs[0] is deleted
    end if
  end loop
```

Consequently, we determine whether a regional factor exists using the following algorithm.

Where n is the maximum number of regional histories, α is the threshold for the number of overlapping regions, R is the region, and Rs is a queue of the region history. The length faction returns a number of Rs elements. This algorithm works whenever the user operates a map. We count the number of overlaps of the currently displayed region and previously displayed region. If the number of overlaps is more than the threshold α, we detect the user's strong criteria for that region.

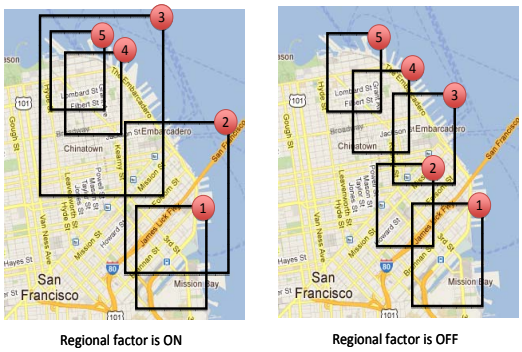


Figure 4: Determining the regional factor

3.2. Extraction of Categorical Factor

Next, we describe how we detect the user's criteria for a category, considering that the user has such criteria. If a user selects a certain category, this category is not always suitable for the implicit criteria. Thus, we use selected objects for detecting the user's criteria for a category. We define the categorical factor, as these criteria can be detected based on an object selection from a category.

When a user selects objects several times and these objects are often included in a certain category, we consider that the user wants to search using this category (see Figure 5). At this time, we detect the categorical factor as the user's categorical criteria. On the other hand, when a user selects objects several times, and these objects are included in widely varying categories, we can consider that the user has no category criteria. Therefore, no categorical factor exists as a categorical criterion.

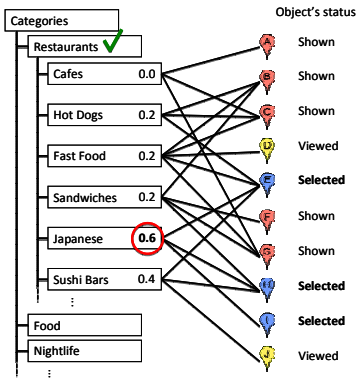


Figure 5: Determining the categorical factor

For the reasons mentioned above, we determine whether the categorical factor exists using the following algorithm.

Algorithm 2 Extraction of categorical factor

```
Require: β = 0.6
  loop
    Waiting for object viewing or object selection
    Categorical factor is OFF
```

```
    Selected object's categories are added to Cs
    if Inputted operation is object viewing then
      Viewed object is added to Vs
    else if Inputted operation is object selection then
      selected object is added Os
    end if
    for i = 0 to Length(Call) do
      for j = 0 to Length(Cs) do
        if Call[i] = Cs[j] then
          Count of Call[i] ← Count of Call[i] + 1
      end if
    end for
    for i = 0 to Length(Call) do
      Ratio of Call[i] ← Count of Call[i] / Length(Vs)
      if Ratio of Call[i] > β then
        Categorical factor is ON
      end if
    end for
  end loop
```

where β is the threshold of the ratio of the selected object categories, Cs is a set of selected object categories, and Os is a set of selected objects. C_{all} refers to all categories. When an object is selected by the user, we calculate the ratio of selected object categories based on Os. If the ratio of the selected object categories is more than threshold β, we detect the user's strong criteria for these categories.

3.3. Extraction of Objective Factor

Object selection means that the user wants to search around the location of a selected object and is interested in the selected object's category. We consider that user's criteria have changed based on the state of the object selection: selected object, viewed object, and shown object. Therefore, we define the three types of objective factors as SELECT, VIEW, and NONE (see Figure 6).

A selected object has both strong region and category criteria. The user selects this object based on its position and information. We define SELECT as including at least one selected object in the current region and selected categories. A viewed object has strong region criteria, however the categorical criteria are weak. At this time, the user selects this object from the map, and does not select it from the object information. In other words, the user agrees with the object's location, and disagrees with its information. We define VIEW as including at least one viewed object in the current region and selected categories without selected objects. A shown object has no criteria beyond the current region and selected categories. We define NONE as not including any selected or viewed objects in the current region and selected categories.

We confirmed following results by preliminary experiment. When participants had selected a regional factor with strong criteria, they did not change the region very often. Because regional factor was more important to them than categorical factor, they did not change the region greatly by often changing categories from a category list. When participants had selected a categorical factor with strong criteria, the frequency of viewing and selecting objects was much higher. Because categorical factor was more important than regional factor, they compared it between objects enough by viewing and selecting at the object belonging to the categories.

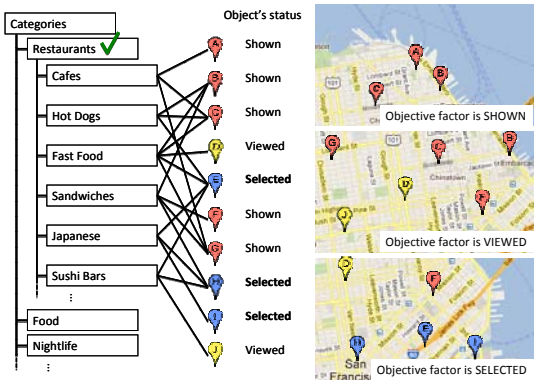


Figure 6: Determining the objective factors

4. Recommendation using Search Condition

4.1. Estimation of Search Condition

We estimate the user's search condition using the regional, categorical, and objective factors. The user's search condition consists of regional and categorical search conditions. The regional search condition is the degree of criteria for the current region. If the regional search condition is high, the user wants to search for geographical objects in the current region. On the other hand, if the regional search condition is low, the user does not want to search in the current region necessarily. Similarly, the categorical search condition is the degree of criteria for the current categories. If the categorical search condition is high, the user wants to search for geographical objects that are included in the current categories. On the other hand, if the categorical search condition is low, the user does not search using current categories necessarily.

Under a constraint such as the number of displayed objects on a mobile device, the search condition is a trade-off problem. We define this constraint as the number of displayed objects being between γ and δ. We use both values of the regional and categorical search conditions for this trade-off. We translate from regional, categorical, and objective factors to values of regional search conditions (R_value) and categorical search conditions (C_value) using the translation rule shown in Table 1.

In Table 1, if the factor has a strong criterion, this value is 1. On the other hand, when the factor has a weak

criterion, this value is -1. If the factor is not related to the search condition, this value is 0. Finally, we add each value from each factor as the regional and categorical search conditions. Therefore, the value of the search condition is between 2 and -2. A value of 2 indicates the strongest criterion, and in such case we consider that the user does not want to change the search property. We describe how we adjust the individual search conditions using the changing region function and changing category function in sections 4.2.1 and 4.2.2. We give an overview of the adjustment strategy in the following paragraph.

We adjust the search property when we detect a trigger condition or operation. Trigger conditions are defined as one of the following. 1) the regional factor is ON, 2) the categorical factor is ON, or 3) the objective factor is SELECT or VIEW. Trigger operations are defined by the number of shown objects after an operation as follows. 1) The number of shown objects is over threshold γ, or 2) the number of shown objects is under threshold δ. we detect a trigger for adjusting a search property when the regional factor is ON and the number of shown objects is more than threshold γ after a panning operation.

Table 1: Translating rule from each factor to each search condition

Factors		*R_value*	*C_value*
Regional factor	ON	1	0
	OFF	-1	0
Categorical factor	ON	0	1
	OFF	0	-1
Objective factor	SELECT	1	1
	VIEW	1	-1
	NONE	0	0

Our proposed algorithm is described as follows:

We adjust the search property using the difference between each search condition. If the difference between R_value and C_value is 0, we adjust the search properties using the changing region function and changing category function together. When the difference between R_value and C_value is n, we adjust the search properties using the changing category function n times and the changing region function once. On the other hand, when the difference between R_value and C_value is –n, we also adjust the search properties using the changing region function n times and the changing category function once.

Algorithm 3 Recommendation using Search Condition

```
loop
    Waiting for regional input, category selection or object
    selection
    if Regional factor is ON or Categorical factor is ON or
    Objective factor is SELECT or VIEW then
        while Number of shown object > γ or Number of shown
        object < δ do
            D ← R_value – C_value
            if D = 0 then
                Do algorithm 4
                Do algorithm 5
            else if D > 0 then
                if Count = D then
                    Do algorithm 4
                    Count ← 0
                else
                    Do algorithm 5
                end if
            else if D < 0 then
                if (Count x -1) = D then
                    Do algorithm 5
                    Count ← 0
                else
                    Do algorithm 4
                    Count ← Count + 1
                end if
            end if
        end while
    end if
end loop
```

4.2. Optimizing Region and Category using Search Condition

4.2.1. Changing Region Function

Next, we explain how we adjust the search property of a region. A map can add the number of shown objects by widening the display region, or reduce the number of shown geographical objects by narrowing this region (see Figure 7). If the number of shown objects is over the threshold γ, we narrow the display region of the map by one level of scale. If the number of shown objects is under threshold δ, we widen the display region of the map by one level of scale. We can add or reduce the number of objects until it is between γ and δ by repeating this process.

Figure 7: An example of changing region

Algorithm 4 Changing Region Function

```
R ← Current displayed region
O ← Current shown objects
if Length(O) > γ then
    R is narrowed one level of scale
else if Length(O) < δ then
    R is widened one level of scale
end if
```

Consequently, we adjust the search property of a region using the following algorithm, where the length function returns the number of currently shown objects, O.

4.2.2. Changing Category Function

Next, we explain how we adjust the search property of a category. We adjust a category by adding a similar category or removing one dissimilar to the currently selected category (see Figure 8). If the categories include a lot of similar objects, the categories are considered similar. When the number of shown objects is over threshold γ, we remove those objects that are in the category that is most dissimilar with selected category. In other words, we add this category to the list of excluded categories. When the number of shown objects is under threshold δ, we add the category that is most similar with the currently selected category. We can add or reduce the number of objects until the number shown is between γ and δ by repeating this process.

For the reasons mentioned above, we adjust the search property of a category using the following algorithm.

Algorithm 5 Changing Category Function

```
C ← Current selected category
O ← Current shown objects
for i = 0 to Length(Call) do
    Sim_Cobject[i] ← Sim(C, Cobject[i])
end for
if Length(O) > γ then
    for j = 0 to Length(O) do
        if Min_category(Cobject) ∈ O[j]'s categories then
            O[j] is removed from O
        end if
    end for
else if Length(O) < δ then
    for k = 0 to Length(Oall) do
        if Max_category(Cobject) ∈ O[k]'s categories then
            O[k] is added to O
        end if
    end if
end if
```

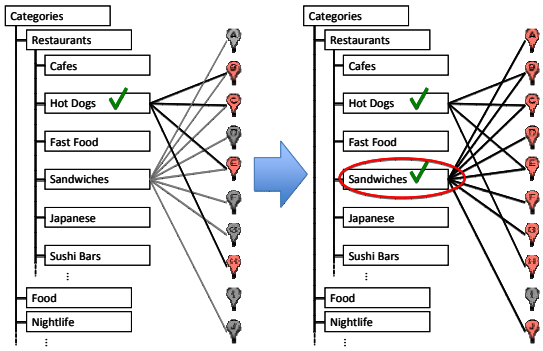


Figure 8: An example of changing category

where the Sim function returns a similarity as a Jaccard index of C_i and C_j's objects. C_{object} is a set of categories that include the shown objects. The Min category function returns a category that has the least similarity, while the Max category function returns a category that has greatest similarity using Sim_C_{object}.

$$Sim(C_i, C_j) = \frac{(\text{Objects in } C_i) \cap (\text{Objects in } C_j)}{(\text{Objects in } C_i) \cup (\text{Objects in } C_j)}$$

5. Prototype System

We developed our prototype system using Java and JavaScript. Our interface was developed using JavaScript as a client system. Figure 9 shows the interface: (a) is a category view in which the user can select or deselect a category, (b) is a map view in which the user can change a region and where the system shows objects included in this region and the selected categories, and (c) is an object list view where the user can view detailed information and select/deselect objects. When a user taps a tab on the upper part of the screen, the system shows a different view. In addition, when a user taps an object icon in map view, the system shows an object list view and detailed information on this object automatically. Our algorithm is implemented through Java as the server system. Our interface sends user operations to the server system. Normally, our system searches for objects using Gurunavi 1 based on user specified properties of the categories and region. When our system determines a trigger using received user operations, our system adjusts the search properties based on the estimated search conditions.

Category view Map view Objects list view

Figure 9: Recommendations using the search conditions of the categories

We set up the parameters as α = 3, β = 0.6, γ = 20, and δ = 5. We performed a preliminary experiment to confirm the relationship between user operations and user criteria for determining α and β. Participants used Gurunavi as a search engine for finding a restaurant and answered a questionnaire regarding the region and categories of interest. We then collected the operation logs, and conducted an analysis from these logs and the questionnaire results.

We performed a second preliminary experiment to confirm a reasonable number of displayed objects for determining γ and δ. There were 9 participants. They viewed a displayed map with an increasing number of object icons of 1 to 30. The participants decided whether the number of displayed objects was reasonable.

6. Evaluation

We compared the objects recommended by our system with the objects displayed by a conventional system. We carried out 12 different tasks on familiar and unfamiliar regions and categories, as we provided 4 tasks for each of the 3 regions. The tasks were to search for lunch in each of three locations using following conditions.

A. Familiar areas, familiar categories

B. Familiar areas, unfamiliar categories

C. Unfamiliar areas, familiar categories

D. Unfamiliar areas, unfamiliar categories

We let the participants operate our system based on each task until a recommendation screen appeared. After a recommended was given, we let the participants select adapted objects from those displayed in the conventional system and from the recommended objects of our system. Moreover, we did not evaluate the reduced objects in the conventional system based on the recommendations of our system as we evaluated the objects after a recommendation was given. The evaluation of the reduced objects is our future work.

Table 2: Experimental Results

	Familiar area			Unfamiliar area		
	C	O	R	C	O	R
P_1	6	6	0.00	3	6	0.50
P_2	3	7	0.57	0	9	1.00
P_3	3	6	0.50	4	6	0.33
P_4	1	6	0.83	0	6	1.00
Ave	-	-	0.47	-	-	0.71

Table 2 showed that the increase degree of selected adapted objects (column R) from conventional method (column C) to our method (column O). P_x shows participants. At the result of unfamiliar area, the increase degree is higher than familiar area's increase ratio. In particularly, participants do not know it where there are a lot of restaurants in unfamiliar areas. Therefore, it is effectiveness for participants that candidates added by our system when there is little number of the displayed restaurants. From these results, when users use our system in unfamiliar areas, we found that the system was particularly effective. In this way, we confirmed the effectiveness of our system. However, some participants answered that it was difficult for them to understand the scaled-up region because they did not know which part of the displayed region was scaled up.

7. Conclusion

In this paper, we presented a geographical information recommender system for mobile devices based on user search operations. We focused on adjusting the search property from the user's search conditions and enabled an intelligent interaction between realizable user operations and the general interface. In addition, we validated the effectiveness of our recommender system in unknown regions. As future work, we have to improve the recommendation algorithm for objects and evaluate the comfort of the operation and interface. Additionally, as other application, our method is able to apply about other data such as travel information except restaurant information. In this case, it is necessary to adapt multiple data such as room type, price, companion, facilities and so on. Moreover, it is important to extract user's point of view from review data belonging to hotel information.

Acknowledgments

This work was supported in part by a joint research project with Micware co., Ltd.

References

[1] A. Aula, R. M. Khan and Z. Guan. How does search behavior change as search becomes more difficult?, in *Proc. of the 28th international conference on Human factors in computing systems*, pp.35–44, 2010.

[2] M. Cho, B. H. Kim, D. K. Jeong, Y-G. Shin and J. Seo. Dynamic query interface for spatial proximity query with degree-of-interest varied by distance to query point. in *Proc. of the 28th international conference on Human factors in computing systems*, pp.693–702, 2010.

[3] L. Martinez, R. M. Rodriguez and M. Espinilla. A Georeferenced Hybrid Recommender System for Restaurants. in *Proc. of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops*, pp.187–190, 2009.

[4] D. Mountain and A. MacFarlane: Geographic information retrieval in a mobile environment: Evaluating the needs of mobile individuals. *Journal of Information Science*, vol.33, no.5, pp.515–530, 2007.

[5] J. Teevan, S. T. Dumais and E. Horvitz. Personalizing search via automated analysis of interests and activities. in *Proc. of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.449–456, 2005.

[6] H. Toda, N. Yasuda, Y. Matsuura and R. Kataoka. Geographic information retrieval to suit immediate surroundings. in *Proc. of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle, Washington, pp.452–455, 2009.

[7] J. Weakliam, M. Bertolotto and D. Wilson. Implicit interaction profiling for recommending spatial content. in *Proc. of the 13th International Symposium of Advances in Geographic Information Systems*, pp.285–294, 2005.