

FINDING GOOD AREAS FOR RENTING APARTMENTS USING APARTMENTS INFORMATION AND USERS' TRAJECTORIES

Ruochen Si^{a,*}, Min Lu^a, Masatoshi Arikawa^a, Yasushi Asami^a and Junko Iwasaki^b

^a Center for Spatial Information Science, The University of Tokyo
5-1-5, Kashiwanoha, Kashiwa City, Chiba 277-8568, Japan - si@csis.u-tokyo.ac.jp
^b At Home Co., Ltd., Japan

KEY WORDS: Apartment Renting, Location Based Services, Ubiquitous Mapping, Moving Trajectory, User Context

ABSTRACT:

Current apartment renting web sites provide services to search apartments by attribute filters while they lack the services to help find areas that are convenient for living and meet customers' personal preference. In many cases, people not only need attributes of the apartments meeting their requirements, but also need the attributes of the area be satisfied. We propose an IT service that can find appropriate areas for customers to rent apartments while they want to move to new places. The areas are picked according to users' life styles mined from their former moving trajectories and to the attributes of the areas calculated from preferences of present rooms of users.

1. INTRODUCTION

Apartment renting has a big market, especially in big cities like Tokyo. There are many web sites collecting a large sum of apartment information. These kinds of services make it convenient for customers to search available apartments. However, to find an ideal apartment is still time consuming and a headache. Although the total amount of information on web sites is large, in many cases the information that users really care for choosing an apartment is limited, especially the information of the areas around the apartments. This makes it difficult for customers to make right decisions. To find a good area to live is important to customers. On one hand, customers usually prefer living in an area that meets their preference. We can imagine that those who are allergic to animal hair would prefer the area with little pets. On the other hand, people also want to live near to the important places, like their schools and working places. However, current web services usually provide mainly attribute filters for specific apartment while they are weak in offering the information and evaluation of areas fitting for customers.

Besides, the information on the websites is usually for the general public instead of for a specific customer. Contexts are used for adapting the services and information offered to a user (Mikalsen and Kofod-Petersen, 2005). Context aware applications can examine and react to an individual's changing context (Schilit, Adams and Want, 1994), and it is needed to offer personalized services. The user context is important and it is researched widely. Mayrhofer et al. introduced an architecture that allows mobile devices to continuously recognize current and anticipate future user context (Mayrhofer, Radi and Ferscha, 2004). Anagnostopoulos et al. presented a software architecture for context awareness in mobile computing environments, sensor centric systems and discuss context modeling issues (Anagnostopoulos, Tsounis and Hadjiefthymiades, 2007).

In this paper, we propose an IT service that can find appropriate areas for specific customers to rent an apartment, using the attribute information of apartments and the customers' moving trajectories. The service we propose mainly serves for a user who has lived in one place for some duration and wants to move to another apartment nearby. For example, a university

student has lived in a school lodge for one year and is going to move out.

The rest of the paper is organized as follows: Section 2 presents a method to pick areas that are good for users to rent apartments; Section 3 introduces an implementation of the method proposed in Section 2 and the simulated test of it; Section 4 draws the conclusion of the paper.

2. METHOD TO PICK HOT AREAS FOR RENTING APARTMENTS

We pick areas that meet user's requirements on two aspects. On one aspect, we evaluate attributes of a local area by the corresponding attributes of the apartments in the area. On the other aspect, we use users' moving trajectories through smartphones, and find users' moving patterns, and then evaluate the areas according to the patterns. Finally, we overlay the areas picked on the above two aspects to obtain the areas of our interests.

2.1 Evaluating Areas by Attributes

Most apartments renting supporting websites just provides attribute information of individual apartments, such as rent and size of the apartments, facilities in the apartments, rules of the apartments, and so on. For some people, some of the attributes are not concerned for individual apartments, but for the areas around the apartments. For example, for those who dislike animals, they usually do not care whether their apartments allow to keep pets, but care more about whether the apartments nearby allow to keep pets.

To calculate the attributes of the areas from the attributes of the apartments, we applied the quadrat statistic method. We use grids to divide the whole research area into small subareas. The value of an attribute of the area is calculated by the corresponding attribute of the apartments that falls into this area. The specific method to work out the attribute of the area varies. For example, the attribute of "keeping pets is allowed" is usually calculated by the total number of apartments that allow to keep pets in the area; while the attribute of "rent" is usually calculated by the average rent of the apartments in the area. Figure 1 shows the method to calculate the attribute value of an

area from the attribute of individual apartments. The red spots represent the apartments (noted as O_m , m is the index number of the apartment) that locate in grid with row i and column j (noted as G_k , k is the index number of the grid). The value of the attribute A of G_k is shown in Equation 1.

$$A_{G_k} = f(A_{0_1}, A_{0_2}, \dots, A_{0_m}, \dots, A_{0_n}) \quad (1)$$

where A_{G_k} = value of attribute A of grid G_k
 A_{0_m} = value of attribute A of apartment O_m in grid G_k
 n = the number of apartments in grid G_k

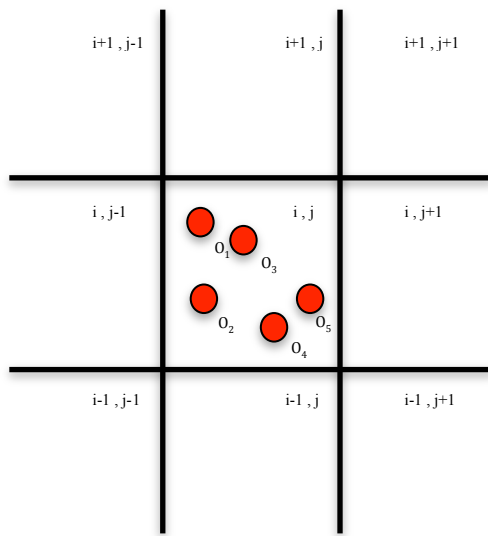


Figure 1. Calculate attribute values of an area from attributes of individual apartments

2.2 Evaluating Areas by Users' Moving Trajectories

The apartment renting websites only have general information of apartments, but not the personalized information for specific customer. We mine the spatial moving patterns and pick the areas that fit their own life styles.

There are places that are relatively more important to other places. People stay there for long time, go there frequently, or go there regularly, such as home, school, working place, transportation stations, and so on. For the task to find good areas to rent apartments, we classify the important places into two categories: (1) unique places, which do not change while people move to a new place, like the school that the user is in; (2) alternative places, which can be replaced by the same kind of places when people move, like the railway stations.

First, we will find the important places for users. We record users' locations with certain time interval. Then overlay users' locations with the grids that have been set up in section 2.1. The grids with more locations falling into are the areas the user often goes to or stays long. These areas are likely to be more important to the user. While the grids with less locations falling into are the areas the user does not often go to, and are usually less important to the user. Then, we judge what kind of place it may be using the location and the timestamps. For example, the grid with a lot of locations and including the timestamps around 3am is likely to be the home of the user. In some cases, users' behaviors may not be regular. For example, some graduate students do not have seats in the lab and do not go to school

regularly. So we also allow users to explicitly select important places for them directly on maps.

Once we find the important places for users, we evaluate the weight of each important place. The weight of an important place (noted as w) is defined to be proportional to the number of user locations falls into the grid here, as shown in Equation 2. Then for each grid, we calculate the weighted distance (noted as Dw) to all the important places in a local area. Equation 3 shows the weighted distance for grid G_i . In Equation 3, the $dist_{i,j}$ is an Euclid distance between the central point of grid G_i and the important grid G_j ; the w_j is the weight of important grid G_j . And as mentioned before, we classified the important places into two categories, and for the alternative places, the distance is from the grid G_i to the nearest grid that contains the same kind of object. The weighted distance indicates the distance the user is likely to go every time when he/she sets off from home. The areas with small weighted distances are more convenient for the user to go to the important places.

$$w_i = \frac{n_i}{\sum_j n_j} \quad (2)$$

$$Dw_i = \sum dist_{i,j} \cdot w_j \quad (3)$$

where n_i = the number of apartments user locations falls into grid G_i

2.3 Integrating Areas Good for Renting Apartments

In the above sections, we have picked the areas from the aspects of attributes of the apartments and moving patterns of the users separately. Here, we are going to combine these two kinds of areas and obtain the final result of the areas that fit for the user to rent the apartment.

It seems that it is easy to integrate the values numerically. For example, integrating by the function $v = w_1 \cdot v_1 + w_2 \cdot v_2$, where v is the integrated value, v_1 and v_2 are the independent values calculated from different aspects, w_1 and w_2 are the weights of v_1 and v_2 . However, simply numerically integrating brings problems.

One problem is that the meanings of the values of each grid area are different when we calculate from the above two aspects. The grid value calculated from the attributes of individual apartments indicates the total number, density, or average value of the attribute of the apartments in the area. While the grid's value calculated from the user's moving patterns indicates the convenience for the user to visit the important places of the user from the grid area. When we integrate the two kinds of values numerically, the meaning of the value will be lost. The simply numerically integrated values can be just used for comparing the grid areas relatively, but we cannot tell the meaning of a grid with a specific value. This problem will make it difficult to set a threshold to tell which areas are good enough for the user to rent apartments.

Another problem is that although in general the grids with higher (or lower) integrated values seems better theoretically, in real cases, such areas may not necessarily satisfy the customers. This problem usually happens in the case that the very high (or very low) integrated value comes from one extremely high value and one extremely low value. It means that the area meets users' requirements on one aspect but fails to meet the requirements on the other aspect. Generally, users do not like this area because they hope the area to meet their requirements on both aspects.

To prevent the above problems, we do not integrate the values numerically. First, we set the threshold for the independent values. Then, we obtain the Boolean result from each independent value of each grid. And finally, we obtain the integrated value by Boolean operations. For example, we note p_1 as the Boolean value for the grid if independent value v_1 meets the requirement, and p_2 as the Boolean value if independent value v_2 meets the user's requirements. Then, the integrated value is an enumerated value in the set $V = \{p_1 \wedge p_2, \neg p_1 \wedge p_2, p_1 \wedge \neg p_2, \neg p_1 \wedge \neg p_2\}$. The areas with the value $v = p_1 \wedge p_2$ represent the areas that both meet the user's requirements on the attribute and also near to the user's important places, and these are the areas we suppose to be good for the user to move to and rent apartments there; the areas with the value $v = \neg p_1 \wedge p_2$ or $v = p_1 \wedge \neg p_2$ meet either requirements for the above aspects; the areas with the value of $v = p_1 \neg \wedge \neg p_2$ meet neither requirements.

3. IMPLEMENTATION AND SIMULATED EXPERIMENT

We realized our proposed method by developing a prototype of an application on Apple Inc.'s iPhone 5, and did simulated experiment to test it.

The data of the apartments for renting is provided by At Home Co. Ltd. The apartments in the data are distributed mainly along Tokyu Railway lines. The data contains properties of apartments such as the ID number, the address, rent, size, whether keeping pets is allowed, whether playing music instruments is allowed, whether only for females, whether only for males, and so on. We obtain the location coordinates of the apartments from their addresses using an address geocoding service offered by the Centre for Spatial Information Science, the University of Tokyo. Figure 2 shows the distribution of the apartments in the data.

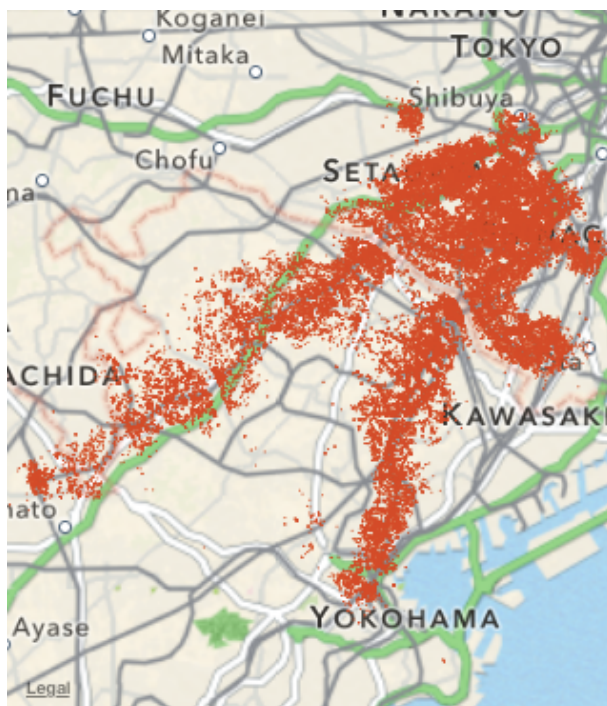


Figure 2. Distribution of the apartments in the data (The red points represent the apartments)

Then, we divided the minimum border rectangle of the researching area into 100m×100m grids. Each grid has an ID

defined by row index and column index, an apartments array that records the ID number of apartments that locate in it, a float variable to record the attribute value that calculated from the apartments, a trajectory array to record user's locations and corresponding timestamps, and another float variable to record its weighted distance to important places.

Next, we overlay the location of the apartments with the grids, and for each grids, we record the ID number of the apartments. Since our purpose is to find good areas for renting apartments, we only keep the grids that contain at least one apartment as the valid grids, and remove the grids that do not contain any apartments from the grids array. Then we are able to calculate the attribute value of the grid area according to the apartments in its apartments array, and tell whether the attribute value of the grid meets the user's requirements with the user input threshold. Figure 3 shows the attribute value of "whether keeping pet is allowed" (this attribute value equals to the number of apartments that allows to keep pets in the grid area).

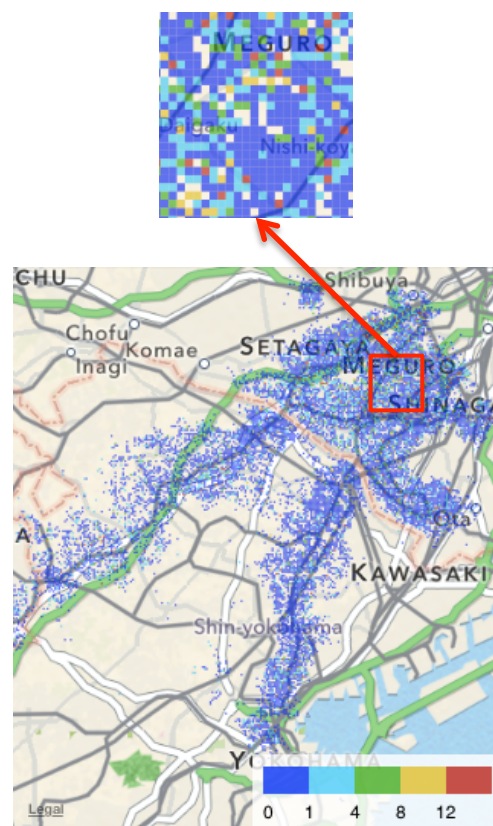


Figure 3. Grids value for the attribute "whether keeping pet is allowed"

Since we do not have the data of the spatial objects on the maps, in this prototype, we input the locations of railway stations manually. Then we let the users to input their homes and schools (or working places). User points the location of the important places on maps those grids that the user pointed in are treated as the important places. And every ten minutes, we detect the user's location with the smartphone, and add the location coordinate and timestamp to the location array of the corresponding grid. In real cases, different parts of a spatial object sometimes fall into different grids. Thus, in our prototype, while calculating the weight of the important places, we count not only the visiting times in the exact grid, but also the grids around it.

Because the prototype is just developed, we do not have enough time to do real test in the real world so far. Instead, we

did simulated experiment. We suppose a user who is going to move and is looking for new apartment to rent. We suppose he does not like animals and hope to live in areas that with little animals. He also does not want to live far from his working place and railway station. We input the working place directly on the maps.

For the attribute of “whether keeping pet is allowed”, we set the requirement as $v = 0$ for the grid area (no apartment allows to keep pets in that area). In Figure 4, the blue areas meet the attribute requirement, while the red areas fail to meet the requirement. And for the requirement for convenient to go to work and railway station, we simulated some user’s locations around the important places, and set the requirement that the weighted distance should be no more than 800m. Figure 5 shows the areas that meet the requirement on weighted distance.

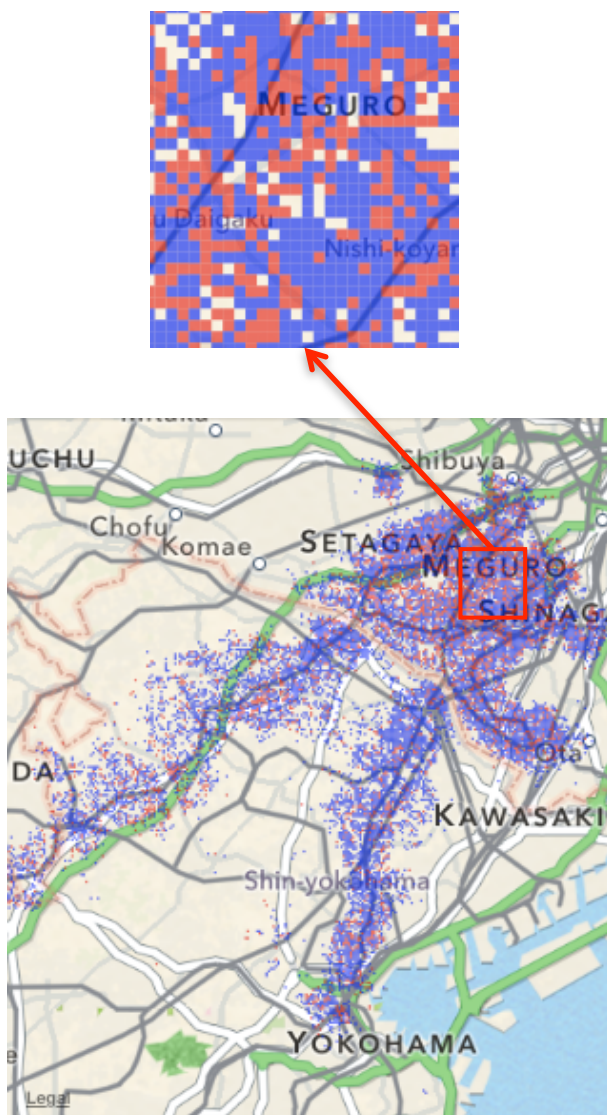


Figure 4. Areas meet the attribute requirements (the blue areas) and areas fail to meet the requirements (the red areas)

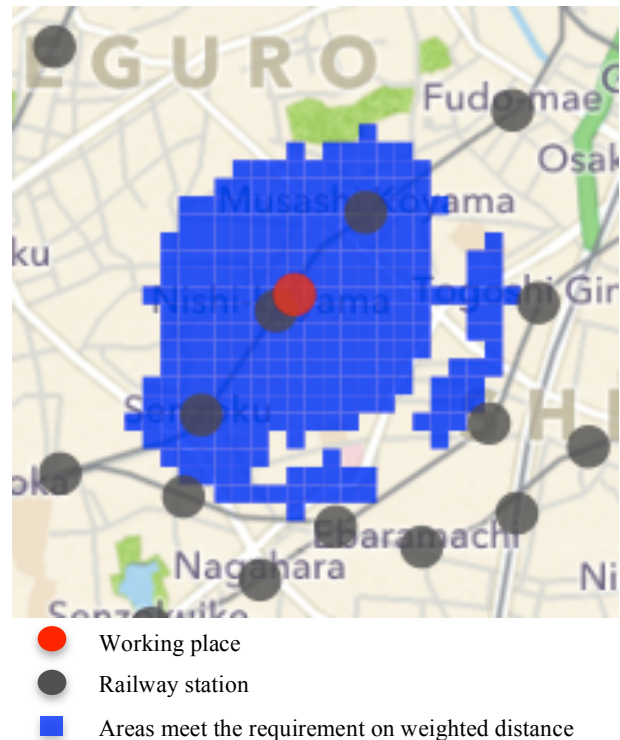


Figure 5. Areas that meet the requirements on weighted distance

While we overlay the above two areas, we obtained the final result, the areas meet both requirements both on attributes and distances to important places. As shown in Figure 6, the blue areas meet the requirements on both attributes and weighted distances, and these areas are recommended to the user for renting apartments there; the green areas meet only the requirement on weighted distance; and the blue areas only meet the requirements on attributes. (The blank areas meet none of the requirements, or do not have apartments for renting.)

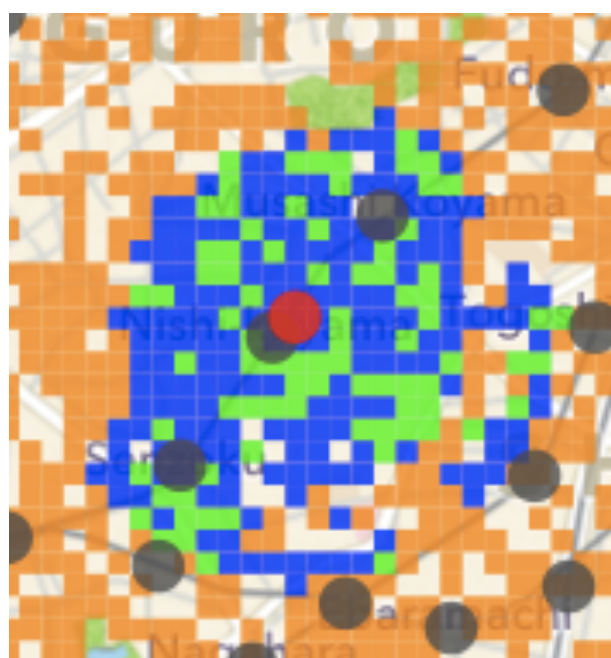
4. CONCLUSION

We proposed a service to help customers to find good areas to rent new apartments when they move out from old apartments, and have developed a prototype application.

The main advantage of this service is that it does not only find areas according to general information of apartments, but also use users’ trajectories to provide the personalized information on the areas that fit users’ moving patterns.

However, our research is still preliminary, and the situation we designed is relatively simple. Now we just consider the moving pattern of individuals but did not consider the moving pattern of group of people, like families. In families, different family members have different important places, and the places that some family members usually visit are treated more importantly than other members. For example, before the kids entering schools, the distance to parents’ working places are considered to be more important while moving; and after the kids entering schools, the distance to the kids schools are considered to be more important.

In the future, we are going to research more complex situations, involving group of people and so on.



- Working place
- Railway station
- Areas meeting the requirement on both attribute and weighted distance
- Areas meeting the requirement only on weighted distance
- Areas meeting the requirement only on attribute

Figure 6. Areas that meet both requirements

References

- Mikalsen, M. and Kofod-Petersen, A., 2005. Representing and reasoning about context in a mobile environment. *REVUE D'INTELLIGENCE ARTIFICIELLE*, 19, pp. 479-498.
- Schilit, B.N., Adams, N.I. and Want, R., 1994. Context-Aware Computing Applications. *Proceedings of Workshop on Mobile Computing Systems and Applications*, pp. 85-90.
- Mayrhofer, R., Radi, H. and Ferscha, A., 2004. Recognizing and predicting context by learning from user behavior. *The International Conference On Advances in Mobile Multimedia (MoMM2003)*, 171, pp. 25-35.
- Anagnostopoulos, C.B., Tsounis, A. and Hadjiefthymiades, S., 2007. Context Awareness in Mobile Computing Environments. *Wireless Personal Communications*, 42(3), pp. 445-464.