Location-Based Agenda using GPS with K-Nearest Neighbor Algorithm Context Matching

Ary Mazharuddin S, Royyana Muslim Ijtihadie, Ivan Ahmed Giovanni, Hudan Studiawan, Waskitho Wibisono, Tohari Ahmad
Department of Informatics, Faculty of Information Technology Institute of Technology
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: ary.shiddiqi@gmail.com

Abstract – Personal reminder application has become widely used for many people to manage their agendas. The most popular reminder are using the calendar-based application. This method will provide notification should an agenda is coming up. Therefore, the user will have a time to prepare for his needs during the activity. However, this method does not support notifications if someone are in the place where the agenda is about to take place. This is because the time is not close enough to give an alert. it would be better if the reminder uses a location-based trigger to remind what should be prepared in order to accomplish an upcoming agenda.

This research developed a combination of Location-based technology service and GPS to provide a context based agenda. This research will record point of interest of users to be related to files that will be used. This technique implements the concept of context-aware with k-NN query or k-Nearest Neighbor classification algorithm to provide suggested files to users.

Experiment results show that the method is accurate in providing the files needed for an agenda. This research suggest that the use of k-Nearest Neighbor algorithm with k = 7 in the context-aware technique provides the best result of the K-NN query.

Keywords: Context-aware, GPS, k-Nearest neighbor.

I. INTRODUCTION

Having a reminder of our agenda has become an important issue for ones with a bunch of activities. Most people synchronises their agendas on mobile phones. The calendar on the mobile phone is used as a trigger when an agenda is coming up. People will set the time to remind for a particular agenda. Usually they tend to set the reminder close enough to the date, so they will have a time to make some preparation. This type of reminder has the weakness that if someone is at the place where a certain agenda in the future will take place at the same location, the reminder will not notify the user. Sometimes, this will bring a disadvantages, because the person should have more time to prepare things that will be needed for a certain agenda in the future. This method had been experimented by Tze Ming Hung et. al. however, it only provide only reminder for an agenda [1].

The current Android platform has been widely used for location service experiment. The presence of location Sensor technology such as GPS (Global Positioning System) and GLONASS attached on Android platform can be used to overcome the above problems [2, 3]. The technology can be combined with other technologies to satisfy the need. For example, the LBS (Location Based Service) technology which is also commonly used in smart phones [4]. The LBS system works by using GPS sensor to locate user's presence. The LBS benefit will be increased if it is combined with context-aware system [5]. To elaborate the mechanism, a Context-aware system is used to give suggestions about what context are we in, as well as what equipment should we carry on the context. This research developed a system that implements context-aware system using k-NN [6] query algorithm used as a classification algorithm to match the current context with context database stored in the system.

a) Context-Aware System
Context is information to describe the situation of an entity. The Entity can be a person, place, or object which is considered relevant to an activity in terms of location, time, activity, and preferences [5, 7]. A system is said to be context aware, if the system can extract, interpret, and use information from the context. In addition, the system can also adapt to the context that is used [5]. In other words, a system is said to be context-aware if the system uses context to provide relevant information to the user, in where relevance dependent on user activity.

b) GPS (Global Positioning System)

GPS is a satellite that provides navigation system which was first used developed by the United States Department of Defense [2]. Through GPS, a user is able to acquire geo-position (latitude, longitude, and altitude) which is measured from sea level. The GPS uses satellites that surrounds the world, where each satellite covers a certain area of the world. GPS transceivers transmits and receives signals produced by GPS satellites. In determining the position, it takes at least three satellites to locate a particular spot in two dimensional view (latitude and longitude), and four satellites for determining three-dimensional view (latitude, longitude, and altitude). The more satellites involved in determining a position, the more accuracy we will get. To obtain maximum signals receive, the GPS devices must be placed in an open space. The presence of barriers, such as buildings, tunnels, etc. effects the signal receive, hence the determination of position will be reduced in its accuracy.

c) Haversine Formula

Haversine formula is the formula used to determine the distance between two points in the world measured from their longitude and latitude [8]. Equations 1, 2, and 3 are two-point distance measurement formula. With the use of latitude and longitude points A and B, their distance is the calculated. $\Delta\phi$ is latitude difference $\Delta\lambda$ is longitude difference. While $\phi_1$ is the first latitude $\phi_2$ is the second latitude, and $R$ is the radius of earth, that is a 6,371 km.

\[
a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1).\cos(\phi_2).\sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (1)
\]

\[
c = 2.\arctan(\sqrt{a},\sqrt{1-a}) \quad (2)
\]

\[
d = R \cdot c \quad (3)
\]

The calculation will produce a number in degree unit. This is because the distance between the two locations is not always a straight line due to the round surface of the earth.

d) K-Nearest Neighbors Classification

K - Nearest Neighbor (KNN) algorithm is a method to classify objects based on the closest matching entries obtained from training data. Training data has several attributes that represents its characteristics, this is modeled as many-dimensional space representation [9]. The space is divided into sections based on the classification of training data. An entry in this space is marked if the class c is the entry with the closest entry with in a K nearest neighbors range. The distance between neighbors are commonly calculated using Euclidean distance and Manhattan distance.

In the training phase, the algorithm stores unique features represented as vectors and the class of each data. In the classification phase, the whole distance of training data are calculated against an object to be classified. Once the result obtained, the distances are then sorted ascendingly to obtain the closest to the farthest similarity. Then, from the sorted data, it is selected in a number of K to obtain K number of stores entries with the closest distances.

![Figure 1. Illustration of k-Nearest Neighbours](image-url)
The higher the value of K will reduce the effect of noise on the classification. However, this will cause the boundaries between each classification becomes more blurred. An example of unknown object class is shown in Figure 1. A object is about to be classified using K-NN algorithm. If we set K = 3, the object is classified into triangular class because 2 of 3 objects in the K=3 boundary are triangle. However, if the number of K is increased to be K = 5, the objects is classified into class star shape because there are more star that triangle in the K=5 boundary.

Mathematically, majority voting method in the K-Nearest neighbor can be formulated as follows [9]:

Majority voting:

\[ y' = \arg\max_v \sum_{(x,y) \in D_z} I (v = y_i) \]  \hspace{1cm} (4)

**Input:** D dataset training object
**test data object:**

\[ z = (x', y') \]  \hspace{1cm} (5)

**Process:**
Calculate \(d(x', x)\), The distance between \(z\) and each object, \((x, y) \in D\) choose \(D_z \subseteq D\). Data set of K objects nearest training of \(z\).

**Output:**

\[ y' = \arg\max_v \sum_{(x,y) \in D_z} I (v = y_i) \]  \hspace{1cm} (6)

Where \(v\) is the class label, \(y_i\) is the class label of the i-th nearest neighbor and I is an indicator function that returns a value of 1 if the argument is right and 0 if it is wrong.

B. SYSTEM DESIGN

Figure 2. Software architecture of the system.

The system architecture is illustrated in Figure 2. The sequence of the process will be explained as follows: The process of point of interest data classification starts when the application initiated. Then, the application will look for the position of the user by contacting GPS to obtain user’s longitude and latitude. When the user’s position is acquired, the system is then checks if there’s an activities at the current position within a radius of one kilometer. Should there’s an activity in the future at the current location, then a point of interest enquiries is performed by sending the user latitude and longitude as well as the context of the task at the time. A point of interest classification process is done at
the server side with regard to the distance to user’s future activity location and contexts according to the classification. After completion of the calculation, then the point of interest data is transmitted i.e. the location of the future user activity to the user’s current position.

Figure 3 shows the process performed by the system. There are several functions that need data from mobile phone sensors such as getting location data from the GPS sensor. While other functions require data from SQLite database such as tasks/activities data that has been stored on cell phones, and also services that need internet connection such as retrieving related data locations from web server. The classification process will produce two types of classes, i.e. main recommendation and no recommendation. It aims to classify the current user position whether there’s a related location to an activity or there’s no location related to it.

Figure 3. Process Diagram of the System

Figure 4. Workflow of K-Nearest Neighbours Algorithm

Figure 4 shows classification process using the k-Nearest neighbor to be used in point of interest classification system. Here, point of interest locations are separated into classes defined by systems at the number of k. The input of the classification process is the value of x, y, and the context of the task as a results of data obtained from the GPS. This process is called data testing. The classification process begins by calculating the distance between \((x, y)\) with each data training \((x_t, y_t)\) using Haversine formula.

A number of data training k with the nearest distance with data testing will be stored. Recall that there will be two classifications results, namely (MAIN RECOMMENDATIONS, and NOT RECOMMENDED) for each data training, the determination of point of interest are based on the two classifications results. An example of the result of the k-Nearest Neighbor classification using \(k = 3\) would result in three category names, they are : 1 = MAIN RECOMMENDATIONS, 2 = MAIN RECOMMENDATIONS location, 3 = NOT RECOMMENDED. By considering the result, the user are suggested to choose the first two results.

When a current location has correlation with activities locations stored in the context database, the system will recommend files related to a particular activity, then the files can be downloaded to be used. Should there’s additional files that will be used in a particular activity, the file dan be tagged with the desired context and then stored in the
database as shown in Figure 5. The new file(s) added will be shown in the similar recommendation provided by the system.

![Related Files Provided for a Particular Context](image)

Figure 5. Related Files Provided for a Particular Context

III. **EXPERIMENT**

In this section, the software is tested to evaluate its validity and performance. There will be several scenarios used, and explained as follows

a) **Experiment Environment**

For this purpose, a customized environment is set to fulfill the need of the system. In this experiment, the specification of hardware is shown in Table 1.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Processor</th>
<th>1.4 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>512 MB RAM</td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>3.7 inch</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brand</td>
<td>Samsung Galaxy Wonder</td>
</tr>
</tbody>
</table>

Table 2 shows a dataset example from the point of interest locations taken from experiment location (ITS campus) which is then used for classification with k-Nearest Neighbor algorithm.

<table>
<thead>
<tr>
<th>latitude</th>
<th>longitude</th>
<th>location_name</th>
<th>Location_context</th>
</tr>
</thead>
<tbody>
<tr>
<td>-7.280523</td>
<td>112.793214</td>
<td>FTSP ITS</td>
<td>lecture</td>
</tr>
<tr>
<td>-7.281446</td>
<td>112.794504</td>
<td>ITS rector</td>
<td>work</td>
</tr>
<tr>
<td>-7.282263</td>
<td>112.791599</td>
<td>Mosque ITS</td>
<td>worship</td>
</tr>
</tbody>
</table>

b) **Functionality Experiment**

An experiment was conducted to observe the validity of the system. The experiment was aimed to test the functionalities of the application features. Table 3 shows the data of functionality experiment.

<table>
<thead>
<tr>
<th>No.</th>
<th>Name of Experiment</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User location detection test</td>
<td>Succeed</td>
</tr>
<tr>
<td>2</td>
<td>Test checks the user task</td>
<td>Succeed</td>
</tr>
<tr>
<td>3</td>
<td>Test checks related locations on the map</td>
<td>Succeed</td>
</tr>
<tr>
<td>4</td>
<td>Test classification related to the location of various ( k ) values</td>
<td>Succeed</td>
</tr>
<tr>
<td>5</td>
<td>Trial to view and download the files associated with the task</td>
<td>Succeed</td>
</tr>
</tbody>
</table>
Experiment result in Table 3 shows that all the features on the system works well. This can be observed from the success rate of the GPS sensor and context-aware components.

c) Performance Experiments

This experiment scenario is divided into two parts, namely the accuracy and processing time testing. The details of both experiments are explained as follows.

- Accuracy Testing

This testing was conducted to observe the accuracy of the system in giving suggestions to the user from location classification results based on the context of the activity by alternating the value of variable K in K-NN algorithm.

There are several activities observed, i.e. sitting, walking and driving. The test results is shown in Figure 6. The Figure shows that the accuracy increased when K value was raised from $K = 3$ $K = 5$ and $K = 7$. It can be concluded that the best value of $K$ are $K = 5$ and $K = 7$ with the accuracy of 70% - 80%.

- Processing Time Testing

This experiment is aimed to observe the speed test of the classification system in relation with the location of the user activity. In order to gain maximum result, the experiment used HSDPA connection. By varying k value and dataset used are 132 datasets, the result is shown in Table 3.

<table>
<thead>
<tr>
<th>Value of k</th>
<th>Response time classification (In seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1.80</td>
</tr>
<tr>
<td>5</td>
<td>3:24</td>
</tr>
<tr>
<td>7</td>
<td>3.64</td>
</tr>
</tbody>
</table>
Figure 7 shows the speed test result of the classification process based on the value of k. The measurement starts with the number k = 3 to k = 7. The Results of the time measurement is presented in units of seconds. The response time is measured from the time the classification starts and ends when the response is received by the user. The Figure shows that the increase of K value will cost more delay in the response time. this is because the system will have to find more entries to be suggested to user. However, the increase of the K value will provide more accuracy as explained in the previous section.

IV. SUMMARY

Based on the experiment conducted, it can be concluded that this system is feasible to implement. In addition, with the use of K-NN algorithm, the system provides satisfactory accuracy (70%-80%) that will help the users in organizing their activities. This accuracy is expected to increase if the system is provided with larger data training. The higher the value of K, the classification will provide a more precise result. Another advantage is that this system is light, therefore it will not burden the smart phones that run this system.

IV. ACKNOWLEDGEMENT

We would like to thank to Directorate General Of Higher Education (DIKTI) for contributing in this research by profiding research fund. This research is funded by State University Operational Support (BOPTN) year 2013. This paper is produced as the outcome of the Lab-Based Education (LBE) BOPTN research scheme.

V. REFERENCES