Radiant Detector System: Indoor Localization By Nearby Wi-Fi Routers

Hoon Namkung
Department of Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
Republic of Korea
hooni702@kaist.ac.kr

Jeongho Jeong
Department of Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
Republic of Korea
fiil12@kaist.ac.kr

Sanghyun Kim
Department of Physics
Korea Advanced Institute of Science and Technology (KAIST)
Republic of Korea
harry_ksh@kaist.ac.kr

***Abstract*—We present the RDS (Radiant Detecting System), a system using Wi-Fi lists and machine learning algorithm to realize the indoor localization of restaurants in Eoeun-dong, Daejeon. Our approach uses the fact that many restaurants have their own routers emitting Wi-Fi signals of unique lists and strengths, to differentiate among restaurants that are so close to one another that GPS is hardly applicable.**

Keywords—indoor localization; Wi-Fi

I. Introduction

'Indoor localization' refers to the pin-spotting of a location of a user inside of a building or infrastructure. Many techniques are already being used, the most famous being the Global Positioning System, using multiple satellite signals to solve for the location of an object. Since the past approaches such as GPS cost a lot of resources and efforts, we came to use something more accessible, but no less accurate: our smartphones. Among approaches using smartphones is *SurroundSense* by Duke University team [1]. It utilized various sensors in smartphones to track down the location; while its accuracy was noteworthy, *SurroundSense* uses too much batteries, takes a lot of time for completing the database for location fingerprinting, and is simply outdated considering the rapid advances in smartphone technologies and surrounding environments.

With growing number of routers and widespread installation of Wi-Fi, the possibility of differentiating places with Wi-Fi lists is ever more promising.

We present our Radiant Detecting System (RDS) that remedied shortcomings of the past approaches and fit better to time and geographical background of today. RDS has the following strengths.

First, it is more energy efficient. The only source RDS references to for localization is Wi-Fi lists. In comparison, *SurroundSense* uses four sensors: accelerometer, audio, camera, and Wi-Fi. Naturally, users don't need to worry about battery running out.

Second, the time and effort spent for making the comparison set is reduced, since we only needed Wi-Fi lists; the system got lighter.

Third, despite the relative lightness of the system, the accuracy of RDS has become higher, since it makes an active use of current environment that abounds in the number of routers and Wi-Fis. A thin separating wall is enough for localizing different restaurants, since the strengths of signals undergo changes.

Fourth, with the application of machine learning algorithm, our RDS system is more stable and resistant to anomalies and errors.

There were some challenges to be tackled as well.

First, the lists of Wi-Fi for analysis vary from place to place even within the same restaurant. So there is not just one list of Wi-Fi that can be used to track down the place of a user.

Second, some signals that serve as key factors in list of Wi-Fi s may go offline or missing, or even prove to be signals from cell phone hotspots.

We tackled the first problem by collecting data from at least three different spots in the restaurants. We applied a statistical approach from solving the latter issue. By counting the appearances of Wi-Fi signals in the list, we tried to figure out seemingly-important-but-unreliable signals, and completed our trained set.

II. Background

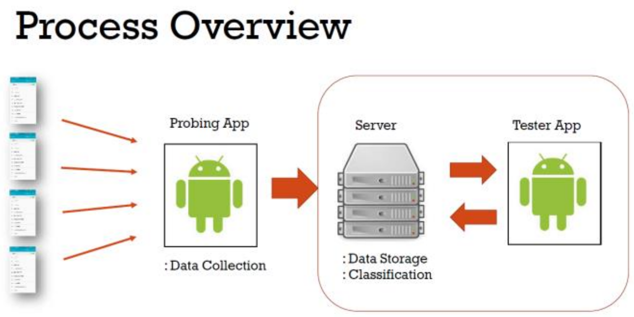
A. Machine Learning

Machine Learning is a field of computer science that deals with pattern recognition and computational learning theory in artificial intelligence [2]. It is applied to three main types of problems: classification, regression and clustering. The first type is about processing input data by labeling them to specific classes; and it is different from regression problem in that the output data is discrete. Clustering problem is about grouping input data into clusters by the similarity among data, and is differentiated from the first two problems in that the clustered groups are not known beforehand, thus being an unsupervised task.

B. KNN Algorithm

K-nearest neighbor algorithm is a non-parametric algorithm used in machine learning for classification and regression [3]. *k* is a user-defined constant for setting the number of existing samples to compare. In KNN classification, the output is a class membership. That is, a sample is classified by a majority vote of its neighboring samples, with the sample being assigned to the class most common among its k nearest neighbors. The process of voting is commonly practiced by calculating the Euclidean distance from the sample of interest to the rest of the comparing set. The main drawback of this majority voting is that more frequent class tend to dominate the prediction of a new sample, since it has higher chance of being near to it.

III. System Design



A. Probing Application

We developed an Android application for scrounging data from Wi-Fi lists in Eoeun-dong restaurants. The app has spaces for writing the restaurants' name, description for the location for data collection, and buttons for starting and sending. The data can be sent to the server real-time or saved in the device first and then be sent. Upon pressing the start button, a new view displaying all the Wi-Fi signals pops out; the list is updated real-time and saved automatically for every three seconds. By touching stop button at the top, the data is saved locally in the device. The data collected by this probing app is crucial both for making a training set and testing the localization system.

B. Server

RDS server uses Amazon Web Service as its source. The language used is Go and the program used for data handling is MySQL. Its main uses are data storing and classifying. Server harnesses the open source library for machine learning developed in Go called 'Go Learn' which we used for classifying our sample data. Server provides a RESTful API, so that anybody can query their location. The server will simply return the logical name of the place based on the given Wi-Fi list, calculated by KNN classifier. The specification of the API is available in our GitHub page. (https://github.com/cs442-radiant/radiant-server) The client can not only query their location but also post collected Wi-Fi list in the same way we did with our probing application.

To use the KNN algorithm, we had to make some choices from our data. There were other information in the Wi-Fi lists other than BSSID and signal strengths: router name, encryption info and etc. We decided, however, to use BSSID and signal strength only, for they were mostly fixed and trustworthy, hence making our process more efficient and powerful.

We set 817 unique routers gathered from 46 restaurants as our vectors for algorithm, and the magnitudes of each vectors were determined by their signal strengths. Instead of deciding the optimum value of K, we discretionally chose K to be 10, 15, 20, 25, 30 since we were not covering a large area and were aware of the number of routers for analysis. However, there may be an issue about scalability, since larger area will have more routers. This case, we can subdivide areas by GPS and run the algorithm per each the divided sectors to solve the problem.

C. Web Page

RDS has its own web page for data displaying and statistical analysis. Its main page has two categories 'database' and 'about us'. Database displays the map of Eoeun-dong restaurants we investigated and names of those restaurants below. Upon clicking the restaurants, bundles are presented. Bundle is a package of samples in the unit of 'save'. That is, all the samples that have been collected after pressing the start button and before pressing the save button are saved in one bundle. Below one level from bundle, the sample data are presented with statistics showing SSID, strength of signal, and number of appearances.

D. Testing Application

This simple testing application is a real-time location checker that utilizes our RDS system. Upon touching the radar image button, it sends the list of Wi-Fi of the location as a query to the server, and displays the reply of the server at the below with 'you are [where]'.

For the system design, we conducted a survey among 53 restaurants in Eoeun-dong to figure out current usages of Wi-Fi routers. 45 restaurants (85%) responded that they have their own Wi-Fi router. Also, out of 45 restaurants, 38 restaurants (84%) responded that they did not replace their Wi-Fi router since they opened. The result shows that there is a high probability that a restaurant has its own signature Wi-Fi router, which is expected to make the classification more accurate. Additionally, since most of restaurants hardly replace their Wi-Fi routers, our system would be robust for a long time period.

IV. Collecting Data Using Probing App

A. Collecting Data Methodology

We designed an application for probing as explained in the previous section. We went out to Eoeun-dong and gathered Wi-Fi lists from 47 restaurants by using this application. We decided to call a single Wi-Fi list as a sample. Sample contains a list of Wi-Fi signal strength and BSSID (MAC address of wireless router) pair measured at specific time. For each restaurant, we extracted average of 60 samples, grouped into 3 categories based on the location (i. e., near entrance door, middle of restaurant, near wireless router). We decided to call these categories as bundles. A bundle is set of samples measured on same location. To sum up, when we entered into a restaurant, we stood near door for 1 minutes to extract a bundle - 20 samples with time interval of 3 seconds and do the same thing in 2 more random location inside of the restaurant.

By taking such measures, we wanted to verify if the samples within the same bundle, which measured in same location, are similar to one another. We also thought that 3 different bundles give diversity to our data set. Finally, this way was efficient because it only took 3 minutes per restaurants.

B. Data Analysis

For 47 restaurant, we could gather 3026 samples and 148 bundles. The average number of wireless routers for sample is 30. The total number of unique wireless routers among 47 restaurants is 817.

As mentioned in introduction, internet penetration rate and mobile usage becomes higher. We thought that average number of 29.7 wireless router at each restaurant is reliable enough to support our system design. Total number of unique 817 wireless router detected from 47 restaurants support the reliability of indoor localization only using Wi-Fi signals.

If we look into samples within a bundle, measured level of signal is almost constant. For overall bundles, average standard deviation is 0.147dBm, which is small enough to conclude overall AP signal strength is almost constant at the fixed position.

TABLE I.

SSID	***BSSID***	***Average Level (dBm)***	***Standard Deviation (dBm)***
KT_WLAN_8CA3	00:27:1c:8c:5a:40	***-71.35***	***7.14***
olleh_WiFi_AB3C	00:07:89:4c:ac:3f	***-78.81***	***7.13***
Olleh_GiGA_WiFi_AB3C	00:07:89:4c:ab:40	***-80.00***	***0.00***
KT_WLAN_8CA3_5GHz	00:27:1c:8c:5a:3f	***-80.77***	***6.24***
	0e:19:70:81:fc:77	***-82.94***	***6.18***

Fig. 1. Top 5 Wi-Fi routers from a bundle measured in *teriyaki hotel*

Among 817 unique routers, many of them are commercial routers that are managed by telecommunication companies. Carrier companies operates free Wi-Fi zone and many of their routers are observed by our system. We found that SK-related router are 103, KT - 102, LG U+ - 65.

This commercial routers are advantageous to our system because more routers gives more fine-grained training data for KNN algorithm so that it helps to increase the accuracy of our restaurant detection. These commercial routers are becoming more prevalent these days, and we can see that our system can fully take advantage of this new environment.

C. Case Study – Eoeun Sushi, Café Crème, Bonjeon Gogitgol

We regard this case as most difficult case since all three restaurants are in a same building. *Bonjeon Gogitgol* is located in second floor, and *Eoeun Sushi* and *Café Crème* are located in the first floor. *Eoeun Sushi* and *Café Crème* have their own signature wireless router whereas *Bonjeon Gogitgol* does not have one.

First, we compared samples between *Eoeun Sushi* and *Café Crème*. Based on average signal strength among 60 samples from *Eoeun Sushi*, we can notice that their signature wireless router is the strongest. Routers from *Café Crème* are also detected, however due to the wall between the restaurant, the strength is reduced. At *Café Crème*, same principle can be applied. *Café Crème*'s signature wireless routers are the strongest, and *Eoeun Sushi*'s signal is reduced. This implies that even though two restaurants are adjacent to each other, our system is able to distinct between two restaurants.

Second, we investigated samples at *Bonjeon Gogitgol* restaurant. Data shows that at *Bonjeon Gogitgol*, Wi-Fi signal from *Eoeun Sushi* and *Café Crème* are detected because they are in the same building, however we can differentiate *Bonjeon Gogitgol* because both of Wi-Fi signal were low.

From this case study, we can conclude that our system have ability to distinct adjacent restaurants. Also, as you can see in *Bonjeon Gogitgol* example, we do not necessarily need signature wireless router for the classification. Nearby routers serve as differentiators for better classification and average 29.7 wireless routers for each restaurant is even more helpful. Many wireless router and differentiators help our system to easily do indoor localization. At the same time, changes for some routers can be regarded as minor issue, because many rest of routers will still remain to cover changes.

Fig. 2. Picture of three adjacent restaurants; Bonjeon Gogitgol, Café Crème and Eoeun Sushi

V. EXPERIMENT EXECUTION & EVALUATION

$$\text{Recall} = \text{true positive}/(\text{true positive} + \text{false negative})$$

$$\text{Precision} = \text{true positive}/(\text{true positive} + \text{false negative})$$

As explained in System Design, we executed our classification algorithm. As we can see, average recall and precision are pretty high - above 90% percent. We can note that the average recall and precision drops as K value increases. The reason lies in our experiment methods. As previously explained in the system design, 50% of the samples were used in testing set, and the rest for training set. One bundle containing 20 samples have 10 samples for testing and the other 10 for training. The reason for high recall and precision for K=10 is that the samples from the same bundle were used for comparison. However, we expected that as we increase the size of the data set, the relation between k and recall/precision will be reversed.

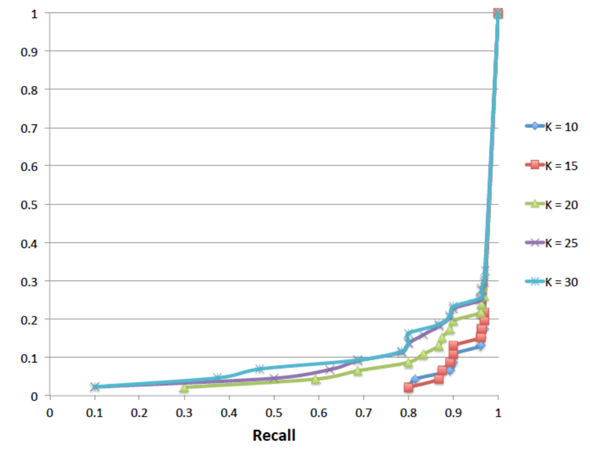


Fig. 3. CDF of recall depending on k value

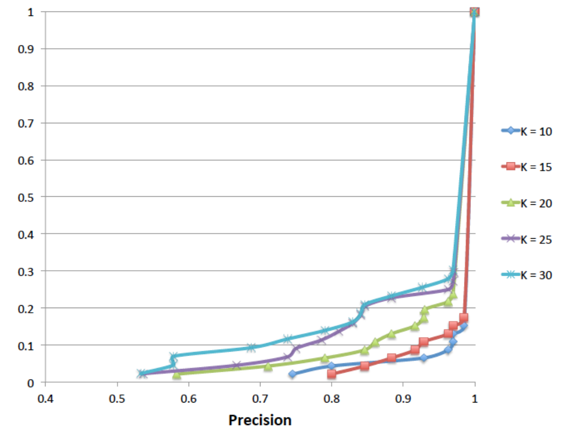


Fig. 4. CDF of precision depending on k value

To prove this viewpoint, we conducted an extra experiment. Low recall means that a restaurant is falsely classified as other restaurant. Low precision means other restaurants are classified as this restaurant. Restaurant with lowest recall is *Hyungjaedolgui*. (recall=0.1 for K=30,25 and 0.3 for K=20) Restaurant with lowest precision is A&A. (precision=0.53 for K=30)

TABLE II.

k	***Average recall***	***Average precision***
10	98.1%	98.6%
15	98.0%	98.4%
20	94.8%	96.4%
25	92.9%	94.5%
30	91.9%	93.4%

Fig. 5. Top 5 Wi-Fi routers from a bundle measured in *teriyaki hotel*

For comparison, A&A and *teriyaki hotel*'s recall-precision values were 1.00-0.67 and 0.5-1.00 for K=25. We thought this case could be revisited to show that the relation between K value and recall/precision can be remedied. We went out to Eoeun-dong to conduct an additional sampling from these restaurants. Unlike previous sampling, we continuously walked around in the restaurants for 2 minutes to collect total of 110 samples from the two restaurants.

Upon running the algorithm, we got A&A's recall-precision value as 0.94-0.81 and *teriyaki hotel*'s value as 0.8-0.93. Likewise, as we expected, larger set of data enabled higher recall-precision. We can conclude that our RDS is expected to work better under provision of larger amount of data.

VI. Conclusion

There has been a variety of efforts for developing more accurate and viable solution for user positioning system. GPS is one of the successful such technologies, but the system is heavy and distinguishing between close places fail from time to time. With the advent of versatile devices such as smartphones and advances in network environment, a new way of indoor localization can be suggested.

In this paper, we present RDS, a new way of indoor localization that is energy efficient, simple and accurate. In this technique, Wi-Fi lists from various routers serve as fingerprints determining the location of the user inside. Our contribution is that we showed the lists of Wi-Fi for localizing actually works well, and adopted machine learning technique to automate the process. We believe that RDS provides a key solution for viable indoor localization that has many future application.

References

[1] Azizyan, Martin, Ionut Constandache, and Romit Roy Choudhury. "SurroundSense: mobile phone localization via ambience fingerprinting." Proceedings of the 15th annual international conference on Mobile computing and networking. ACM, 2009.

[2] Bishop, Christopher M. "Pattern recognition." Machine Learning 128 (2006).

[3] Goldberger, Jacob, et al. "Neighbourhood components analysis." Advances in neural information processing systems. 2004.