Indian Journal of Science and Technology, Vol 8(26), DOI: 10.17485/ijst/2015/v8i26/80494, October 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Smartphone Sensor Value Pattern Analysis with Neural Network

Jaegeol Yim[1*] and Jaehun Joo[2]

[1]Department of Computer Engineering, Dongguk University, Gyeongju - 780-714, Gyeongbuk, Korea; yim@dongguk.ac.kr
[2]School of Management, Dongguk University, Gyeongju, Gyeongbuk - 780-714, Korea; givej@dongguk.ac.kr

Abstract

Location-Based Services (LBSs) have long been studied by many researchers and they nowadays provide very useful information to us, making our daily lives convenient. Vehicle navigation systems are an example of an LBS. As people frequently visit huge unfamiliar buildings, the demand for Indoor LBSs (ILBSs) has rapidly increased. Since indoor positioning is a key technique needed to implement an ILBS system, it has been studied by many researchers. However, a universal solution for indoor positioning has not been found yet. Smartphones are one of the best devices with which interactions between humans and ILBS systems occur because they are equipped with screens, sensors, processors and telecommunications gadgets. Therefore, many research results from smartphone-based indoor positioning studies have been published. One of the most popular methods of indoor positioning is the fingerprint method. The smartphone-based fingerprint method cannot be accurate if the smartphone sensor values collected at one spot are not different from smartphone sensor values collected at other spots. This paper outlines experiments done to show which smartphone sensors can be used in fingerprint indoor positioning.

Keywords: Indoor Positioning, Location-Based Service, Neural Network, Smartphone Sensors

1. Introduction

Location-Based Services (LBSs) such as route services should be available inside buildings and underground structures. Examples of indoor LBSs include museum tour guides and boarding reminders to air passengers far away from their gate[1].

An indoor LBS cannot be realized unless the indoor positioning problem is solved. Because the demand for indoor LBSs has rapidly increased as man-made structures get bigger, indoor positioning has been intensely studied by numerous researchers. Indoor positioning techniques can be classified into two groups; one requires special equipment, whereas the other does not. Lasers, touch sensors, and Radio-Frequency Identification (RFID) readers are examples of special equipment.

The purpose of a Wireless Local Area Network (WLAN) is to provide connectivity to users. People are connected to each other through WLANs and the Internet. An indoor positioning system that uses only WLAN devices can be considered a technique that does not require special devices.

Nowadays, everybody carries a smartphone in order to talk to others, to answer phone calls, to enjoy games, and to connect to the Internet. A smartphone is a kind of daily necessity. Therefore, an indoor positioning system that uses only a smartphone can also be considered a technique that does not require special devices.

There are many indoor positioning techniques that do not require special equipment. The Friis equation–based techniques, fingerprint methods, and dead reckoning are examples that belong to this category. The Friis equation–based techniques make use of signal strength in order to

*Author for correspondence

obtain the distance between a mobile device and an access point. However, the signal strength varies too much without reason, and the estimated positions from this technique are incorrect.

One of the most accurate indoor positioning methods is the fingerprint method. The deployment of the fingerprint method consists of two phases: off-line and on-line phases. During the off-line phase, we build a look-up table. The entire area is covered by a rectangular grid of points called candidate points. At each of the candidate points, we measure the sensor values many times. Let SV_i denote a list of sensor values collected at the i-th candidate point. A row of the look-up table is an ordered pair of ($coordinate_i$, SV_i). A coordinate is an ordered pair of integers (x, y) representing the coordinates of the candidate point.

In the on-line phase, the positioning program gathers sensor values from the user's smartphone. Let Real-time Sensor Values (RSV) denote the sensor values gathered at the moment. If RSV is the same as SV_i in the i-th row of the look-up table, then we conclude that the user is currently located at $coordinate_i$[2].

Therefore, the smartphone-based fingerprint method cannot be accurate if the smartphone sensor values collected at one spot are not different from the smartphone sensor values collected at other spots. Using a neural network, this paper outlines experiments conducted to show which smartphone sensors can be used in fingerprint indoor positioning.

A neural network consists of many computing elements called neurons, and mimics information processing of the human brain. Neural networks are used to solve practical problems in the fields of pattern recognition, system identification, automatic control, and other areas. There are many kinds of neural network. Among them, the Back Propagation (BP) network and the Radial Basis Function (RBF) network are the most popular[3,8].

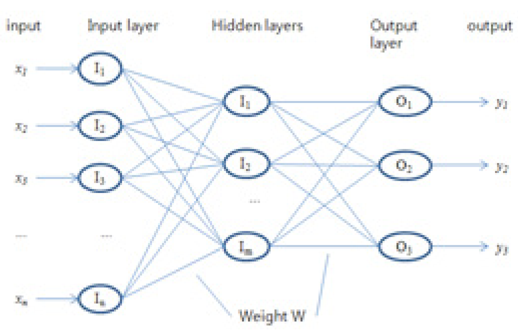

Figure 1. Structure of neural networks.

A BP neural network is a multilayer feed-forward network consisting of one input layer, multiple hidden layers, and one output layer. A layer consists of many nodes. A schematic diagram of a BP neural network with one hidden layer is shown in Figure 1. The learning process in a BP network includes the following steps: 1. Initialize network weights; 2. If there is a training data set, then perform forward propagation and compute any errors, or else exit; 3. Calculate the gradient of the weight, 4. Subtract a ratio of the gradient from the weight, and 5. Go to step 2 if error is greater than a threshold, or else exit[3,8]. MATLAB allows users to easily implement neural network systems.

2. Related Works

Indoor positioning and neural networks are two topics closely related to our work. Liu et al.[3] established back propagation and radial basis function networks with MATLAB to diagnose faults in the subsea blowout preventer[8].

Dheeba and Wiselin[4] proposed a method that discriminates between microcalcification and the normal tissue in the breast with multiple texture features taken from the original image. Feeding the texture features into a multilayer feed forward neural network, they trained the network and tested how well the network classified the input mammogram into suspicious and non-suspicious regions[8].

One of the most significant problems that an air traffic controller has to solve is trajectory prediction. However, trajectory prediction in three dimensions is still not accurate. Sharma and Chaturvedi[5] introduced a feed forward neural network system that can predict an aircraft trajectory in the vertical plane. Taking the current altitude and n previous vertical speeds, the neural network predicts future speed[8].

Vehicle detection is one of the most important problems we have to solve in the development of intelligent transport systems. Among the many techniques used in vehicle detection, Suh et al.[6] implemented a back propagation neural network to detect vehicles. They trained the network with 400 images of the top part of vehicles, another 400 images of the rear part of vehicles, and 130 non-vehicle images. Their test results showed that the prediction accuracy of the network was above 85%[8].

About 20 and 30 kilometers above the earth, the 2-8 kilometer thick ozone layer surrounds the earth. The ozone layer performs the role of protecting living things on the earth from ultraviolet light. This ozone layer has been destroyed by chemicals produced by human activities, such as air-conditioning, spraying pesticides, chemical sterilizing, and cleaning metal parts. Alkasassbeh[7] implemented a neural network that takes nitrogen dioxide concentration, mean temperature and humidity as inputs and generates mean surface ozone concentration as output[8].

Positioning a user is an essential ingredient of a location-based system. For indoor positioning, fingerprint methods are the most promising. Yim[9] introduced a decision tree method where the on-line phase is faster than that of other fingerprint methods.

Yim et al.[10] proposed a relative-interpolation method in order to improve the correctness of outdoor positioning. Then, they implemented an indoor–outdoor positioning method for an LBS on a U-campus with a laptop.

A Kalman filter iteratively predicts the position of a smartphone and updates the predicted position with new measurements. Yim et al.[11] introduced a wireless local area network–based extended Kalman filter method for indoor positioning.

3. Design of the Experiments

3.1 Data Collection

We implemented an Android program that collects smartphone sensor values. Android Studio is now the official Integrated Development Environment (IDE) for the Android operating system. However, we used Eclipse IDE extended with Android Development Tools, which allows developers to use the Android Framework Application Programming Interface (API). In the API, the SensorManager class lets you collect accelerometer, gyroscope, magnetic field, orientation, proximity, rotation vector, gravity, and linear accelerometer values. The LocationManager class allows you to collect information from a Global Positioning System (GPS). The WifiManager class allows you to collect information, including Received Signal Strength Indication (RSSI), from the wireless network card in your smartphone.

The following sentences return an instance of SensorManager, LocationManager and WifiManager, respectively:

mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

Context.getSystemService(Context.LOCATION_SERVICE);

Context.getSystemService(Context.WIFI_SERVICE);

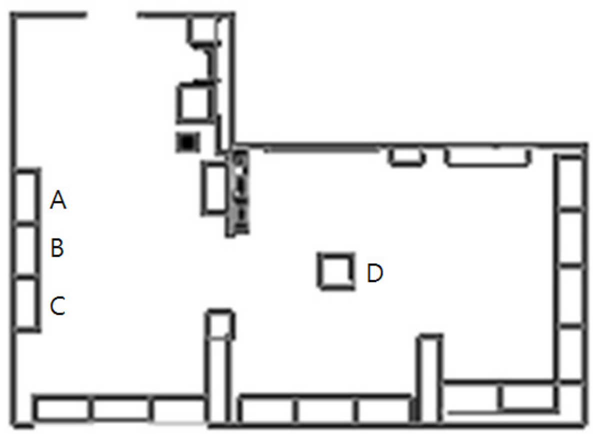

Figure 2. Sensor values were collected at A, B, C and D.

Once we have an instance of the SensorManager class, we obtain instances of sensors as follows:

Accelerometer = mSensorManager.getDefaultSensor (Sensor.TYPE_ACCELEROMETER);

Gyroscope= mSensorManager. getDefaultSensor(Sensor.TYPE_GYROSCOPE);

…

35.8625343	...	35.8625343	...	35.8625343	...	35.8624087	...
129.194836	...	129.194836	...	129.194836	...	129.1947457	...

Figure 3. Part of the collected GPS values.

magnetic X	-4.38	...	-2.22	...	-0.71999997	...	-6.18	...
magnetic Y	-40.86		-42.18		-39.66		-53.219997	
magnetic Z	19.859999		20.58		25.019999		14.639999	

Figure 4. Part of the collected magnetic field values.

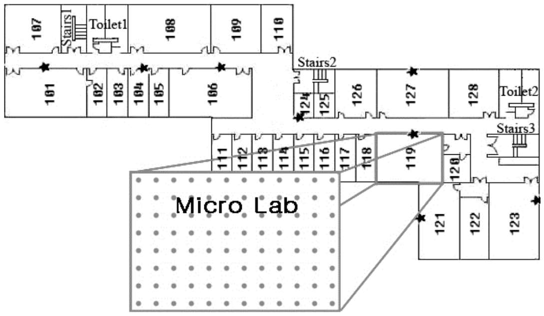

Figure 5. RSSIs were collected at each of the dots shown in this figure.

-50	-51	-51	-51	-51	-51	-52	-52	…
-37	-38	-38	-39	-39	-39	-39	-40	…
-55	-55	-55	-55	-55	-55	-55	-56	…
-77	-79	-79	-79	-79	-79	-79	-79	…
-61	-61	-61	-61	-61	-61	-61	-61	…

Figure 6. Part of the collected RSSIs.

Then, we register sensor listeners as follows:

mSensorManager.registerListener(this, Accelerometer, …);

mSensorManager.registerListener(this, Gyroscope, …);

…

We collected sensor values 10 times at each of the spots denoted by A, B, C and D in Figure 2. The figure represents the configuration of one of the rooms of the university museum. Part of the collected GPS values and magnetic field values are shown in Figures 3 and 4, respectively.

We collected RSSIs 30 times at each of the spots denoted by dots in the Micro Lab shown in Figure 5. Some of the collected RSSIs are shown in Figure 6. RSSI values read by the WifiManager class are negative. The bigger the RSSI value, the stronger the signal.

3.2 Neural Network Construction

We construct neural networks with MATLAB. The MATLAB neural network toolbox allows users to do function fitting, pattern recognition, clustering and time series analysis. Given a series of data points, function fitting finds a mathematical function that has the best fit for them. Pattern recognition assigns a label to a given input value. For example, given the pixel values obtained from an image of the letter "A", it returns the label A. Cluster analysis groups a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups[12]. Time series analysis extracts meaningful information (tomorrow's Dow Jones Industrial Average, for example) from time series data[8].

We perform pattern recognition and classification in order to show which sensor values candidate points can be recognized by. In other words, which sensor can be used in fingerprint indoor positioning? The first step to perform pattern recognition with MATLAB is to type in 'nnstart' at the prompt as follows:

```
>> nnstart
```

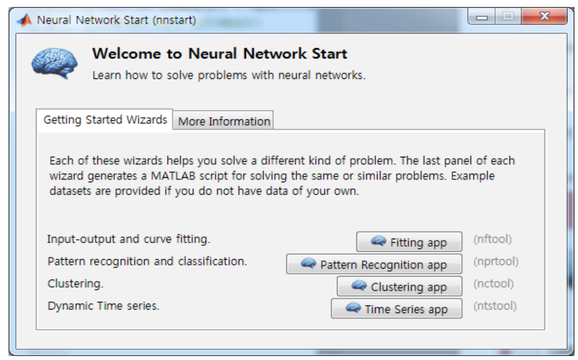

Figure 7. The main graphical user interface of Neural Network Start.

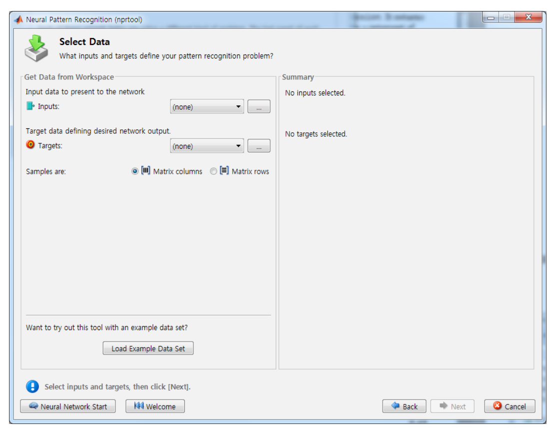

Figure 8. A screenshot of the Select Data page.

The first page of the graphical user interface appears, as shown in Figure 7. Select the "Fitting app" button in order to run the application. It leads us to the "Select Data" page shown in Figure 8 where we can provide both inputs and targets. Inputs consist of a series of data points, whereas the targets consist of desired outputs for all elements in Inputs. For example, we can use the data shown in Figure 3 as Inputs in order to see if the GPS data is suitable to classify candidate points. The table shown in Figure 3 consists of 40 columns: the first 10 columns are GPS values collected at spot A, the second 10 columns represent GPS values collected at spot B, and so on. An example target data for this input is shown in Figure 9. The first 10 columns of the table are (1 0 0 0), indicating that the first 10 columns of the input were obtained at spot A.

1	…	0	…	0	…	0	…
0		1		0		0	
0		0		1		0	
0		0		0		1	

Figure 9. An example of target data for the input data shown in Figure 3.

4. Experiments

The purpose of this paper is to find Android smartphone sensors that can be used in fingerprint indoor positioning. Among the many sensors, we started with the GPS receiver. In other words, we designated the Excel files shown in Figures 3 and 9 as the input and output data, respectively. A screenshot of the Train Network page is shown in Figure 10. In the figure, %E indicates the fraction of samples that are misclassified. The error is greater than 60%. From this training result, we can conclude that the GPS is not a suitable input sensor for fingerprint indoor positioning.

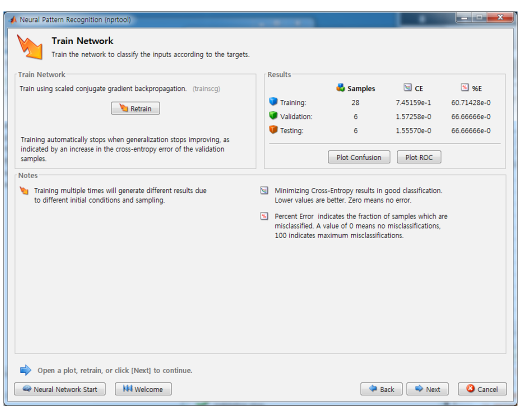

Figure 10. A screenshot of the Train Network page for GPS values.

We performed similar tests with orientation, rotation, gravity, linear accelerometer, and gyroscope, and found that the percent error is above 66%, about 50%, above 50%, above 50%, and above 50%, respectively. With these experiments, we can conclude that these are not suitable sensors for indoor positioning.

However, magnetic field seemed to be suitable input data for indoor positioning. When we changed the number of neurons to 20, we obtained a promising result, as shown in Figure 11. The percent error is not greater than 17%.

After finding that the magnetic field sensor is the best suited sensor for indoor positioning among the smartphone sensors, we continued performing experiments with the RSSIs shown in Figure 6. We can obtain RSSIs using WifiManager. Among the RSSIs, we selected the RSSIs obtained at candidate points 1, 2 and 3 and let them be the input data. A screenshot of the Train Network page obtained by this experiment is shown in Figure 12. It shows that the percent error is less than 9%. After this experiment, we selected the RSSIs obtained at candidate points 1, 3 and 5 and performed similar experiments. A screenshot of the Train Network page obtained by this experiment is shown in Figure 13. It shows that the percent error is 0%.

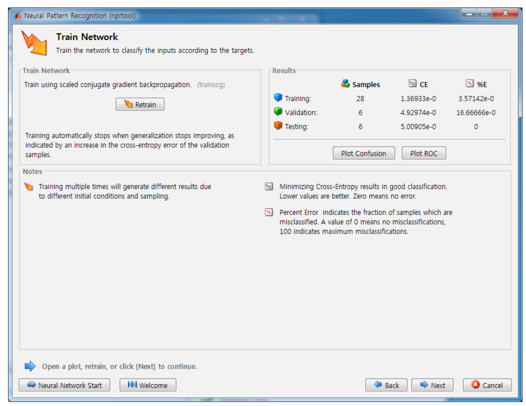

Figure 11. A screenshot of the Train Network page for the magnetic field.

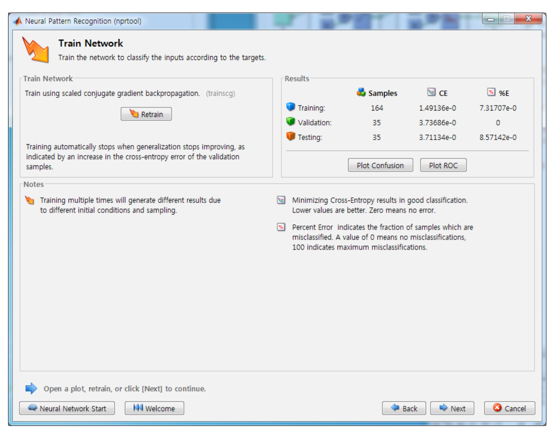

Figure 12. A screenshot of the Train Network page when the distance between two adjacent candidate points is 1 meter.

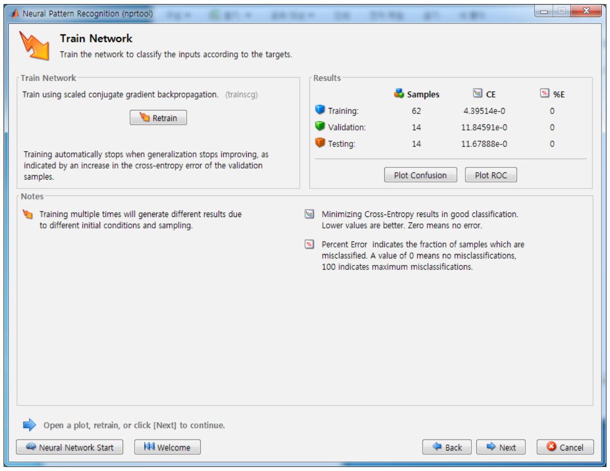

Figure 13. A screenshot of the Train Network page when the distance between two adjacent candidate points is 2 meters.

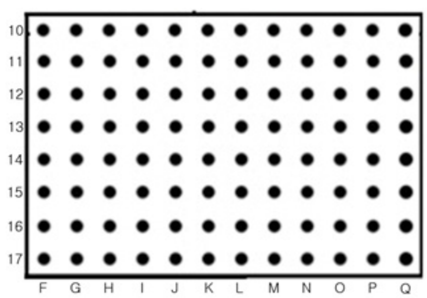

Figure 14. Another test bed.

We performed another set of experiments with the test bed described in Figure 14. We collected RSSIs 30 times at each of the spots determined by an integer from 10 to 17 and an alphabet from F to Q in the figure. Then, we fed in the Excel file consisting of 90 RSSI columns as input data. Among the 90 columns, the first/second/third 30 columns are RSSIs collected at the 10F/10G/10H spots, respectively.

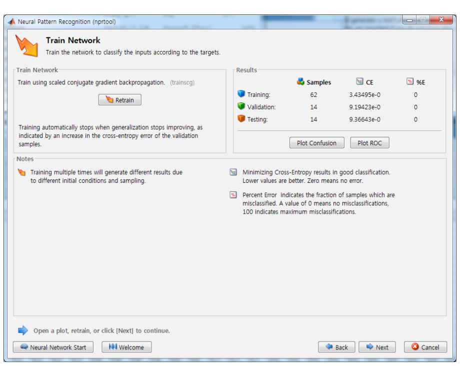

Figure 15. A screenshot of the Train Network page.

Figure 16. A screenshot of the Confusion page.

An example screenshot of the Train Network page is shown in Figure 15. It shows that among the 90 RSSI columns, 62 columns were used for training, 14 columns were used for validation, and the remaining 14 columns were used for testing. It further shows that the percent error was 0. The MATLAB neural network tool also provides confusion matrices, as shown in Figure 16. It shows that the minimum gradient was 0, and it was reached in the 29th epoch.

5. Conclusions

Indoor positioning is one of the key techniques in the development of indoor location-based services. Among the numerous indoor positioning techniques, the fingerprint method is most accurate. The fingerprint method consists of two phases, the off-line phase and the on-line phase. During the off-line phase, the developer has to collect sensor values at certain spots. For example, if the developer collected three sensor values X = (s1, s2, s3) at spot A, then X is considered to be the fingerprint of A. A fingerprint of a person can be used to identify the person. For example, if we have Tom's fingerprint, and it matches the fingerprint found on a coffee mug, then we can conclude that Tom drank the coffee. Similarly, if the sensor values collected by a smartphone some time later are s1, s2, and s3, then the fingerprint method concludes that the current location of the smartphone is spot A.

Therefore, if the sensor values collected at spot A are not different from the ones collected at spot B, then these sensor values cannot be used as the fingerprint. Using the neural network toolbox within MATLAB, we performed experiments to find smartphone sensors that can be used in fingerprint indoor positioning. From our experiments, we concluded that the received signal strength indications collected from the wireless network card can be used as a fingerprint when the distance between two adjacent candidate points is greater than or equal to 1 meter.

6. Acknowledgment

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2011-0006942), by a 'Development of Global Culture and Tourism IPTV Broadcasting Station' project through the Industrial Infrastructure Program for Fundamental Technologies funded by the Ministry of Knowledge Economy (10037393) and the Dongguk University Research Fund of 2015.

7. References

1. Yim J. A Smartphone Indoor Positioning Method. International Journal of Smart Home. 2013; 7(5):9–18.
2. Lee G, Yim J. A review of the techniques for indoor location based service. Internal Journal of Grid and Distributed Computing. 2012; 5(1):1–22.
3. Liu Z, Liu Y, Cai B, Zhang D, Li J. Fault diagnosis of subsea blowout preventer based on artificial neural networks. International Journal of Security and Its Applications. 2014; 8(2):49–58.
4. Dheeba J, Wiselin J. Detection of microcalcification clusters in mammograms using neural network. International Journal of Advanced Science and Technology. 2010; 19:13–22.
5. Sharma A, Chaturvedi A. Gradient descent feed forward neural networks for forecasting the trajectories. International Journal of Advanced Science and Technology. 2011; 34:83–8.
6. Suh H, Kim D, Jang C. Automatic traffic scene analysis using supervised machine learning algorithms - backpropagation neural networks and support vector machines. International Journal of Advanced Science and Technology. 2013; 52:85–92.
7. Alkasassbeh M. Predicting of surface ozone using artificial neural networks and support vector machines. International Journal of Advanced Science and Technology. 2013; 55:1–12.
8. Yim J, Lee Y, Byun J. Implementation of a Neural Network with MATLAB. Proc. The 5th International Conference on Convergence Technology. 2015; 1. p. 780–1.
9. Yim J. Introducing a decision tree-based indoor positioning technique. Expert Systems with Applications. 2008; 34(2):1296–302.
10. Yim J, Ko I, Do J, Jeong S. Implementation of a prototype positioning system for LBS on U-campus. Journal of Universal Computer Science. 2008; 14(14):2381–99.
11. Yim J, Park C, Joo J, Jeong S. Extended Kalman Filter for Wireless LAN Based Indoor Positioning. Decision Support Systems. 2008; 45:960–71.
12. Cluster Analysis. Available from: http://en.wikipedia.org/wiki/Cluster_analysis.