KKU ENGINEERING JOURNAL July – September 2016;43(3):146-152 Research Article

KKU Engineering Journal

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Enhancing indoor positioning based on filter partitioning cascade machine learning models

Shutchon Premchaisawatt and Nararat Ruangchaijatupon*

Department of Electrical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand

Received January 2016
Accepted April 2016

Abstract

This paper proposes a method, called the filter partitioning machine learning classification (FPMLC). It can enhance the accuracy of indoor positioning based on fingerprinting using machine learning algorithms and prominent access points (APs). FPMLC selects limited information from groups of signal strengths and combines a clustering task and a classification task. There are three processes in FPMLC, i.e., feature selection to choose prominent APs, clustering to determine approximated positions, and classification to determine fine positions. This work demonstrates the procedure for FPMLC creation. The results of FPMLC are compared with those of a primitive method using measured data. FPMLC is compared with well-known machine learning classifiers, i.e., Decision Tree, Naive Bayes, and Artificial Neural Networks. The performance comparison is done in terms of accuracy and error distance between classified positions and actual positions. The appropriate number of selected prominent APs and the number of clusters are assigned in the clustering process. The result of this study shows that FPMLC can increase performance for indoor positioning of all classifiers. Additionally, FPMLC is the most optimized model using a Decision Tree as its classifier.

Keywords: Indoor positioning, Machine learning, Wireless device, Filter selection

1. Introduction

Nowadays, the location positioning system becomes increasingly important as it can improve business management or increase convenience in regular life [1]. The Global Positioning System (GPS) technology is widely accepted and used in positioning. However, GPS cannot operate in indoor areas due to various causes, including multipath and signal blockage. Many researchers attempt to invent the new way to position in indoor areas. Several methods, such as triangulation and pseudo GPS [1-4], are proposed. However, none of them are acceptable for indoor positioning in the term of accuracy and cost [1]. Among those proposed methods, one is called the fingerprinting technique, which is more accurate and cost-effective in a real environment [2-3]. The fingerprinting technique collects the received signal strength (RSS) of wireless devices in the indoor area beforehand. Then, the machine learning model is employed to predict the position by relying on the knowledge obtained from the observed RSS data that was collected from the indoor area. However, the performance of classifying depends on training data in the training process. Occasionally, if the collected data cannot provide enough information to classify positions, the result of prediction is unacceptable in terms of accuracy [4].

Practically, there are many access points (APs) in observed locations. These APs can increase performance of positioning by providing more RSS information. However, a large number of APs is not always lead to high performance positioning. In some situation, RSSs from APs can cause miss prediction because noise data is added into the positioning system. The approach to improve performance is increasing information in the system. If the system can find prominent APs that provide informative RSSs, its performance can be increased. In addition, it will also take less time to process.

This research proposes the method called the Filter Partitioning Machine Learning Classifier (FPMLC). FPMLC consists of three processes. The first process is a feature selection that selects prominent APs from several APs. The second and third processes are clustering and classification processes that consist of two cascaded machine learning models for enhancing accuracy. The first model is a clustering model for rough position estimation, i.e. to estimate partitioning areas. The second model is a classifying model to classify a precise position. The performance of FPMLC is compared with conventional methods, such as Decision Tree, Naive Bayes, and Artificial Neural Network. Performance comparison is done in terms of accuracy and error distance. These parameters are widely used as the performance indicators of positioning algorithms.

2. Related works

Several methods for indoor positioning rely on infrastructures and sophisticated hardware. RADAR [5] is

*Corresponding author.
Email address: nararat@kku.ac.th
doi: 10.14456/kkuenj.2016.21

the positioning system by Microsoft that finds positions by using average RSS from many APs. Place lab [6] calculates positions by using average RSS and APs coordinate. Both systems use measured RSS to create the radio map. The machine learning algorithms estimate the location by using the dataset in the radio map [7-9]. Commonly, these methods rely on collected RSS of Wi-Fi AP' reference points in the interested area. The machine learning algorithms learn the relation between RSS and position. Therefore, the machine learning model can predict the position by relying on knowledge which obtained from the training phase. In traditional finger printing method, the standalone conventional machine learning algorithm is used to predict position [1, 2, 4]. Several experimental results provide accuracy and error distance of different algorithms i.e., Naive Bayes [7, 9], Decision Tree [8], and Artificial Neural Network [11]. However, the accuracy of the aforementioned methods is similar with limited accuracy [9]. Some researchers try to enhance the performance of machine learning models for the indoor positioning problem. The example of such research is the Cascade Correlation Networks, which combines two cascaded artificial neural network to improve accuracy [10]. Another one is positioning cascade artificial neural networks, which utilizes space partitioning to increase accuracy [11]. In addition, the fingerprinting method depends on appropriate collected data. In [12], researchers show that appropriate RSS can affect performance of positioning. Consequently, it is necessary to have RSS selection method in order to obtain the finest RSS data.

3. Proposed method

There are many components in the proposed Filter Partitioning Machine Learning. The detail of each components as follows.

3.1 Filter selection

Filter [13] is the algorithm for selecting features; i.e. selecting access points, before process with machine learning algorithms. Filter relies on information gain theory [14], which is used in a decision tree to measure good features for decision making. Filter can determine the prominent access points, which are the access points that provide useful information for positioning. Therefore, the prominent access points lead to correct predictions. Let *D* be the set of all samples that obtained from the measurement. These samples contain relation between RSS from all access points and each position m from all M positions. The number of samples measured at each position m is equal. The information gain of each access point ($gain(ap_i)$) can be calculated by using equation (1).

$$gain(ap_i) = -\sum_{j=1}^{|V|} \frac{|D_{v_j}|}{|D|}\left(-\sum_{m=1}^{|M|} p_m \log_2(p_m)\right) \quad (1)$$

Let *APs* be the set of all access points whose RSS can be measured and ap_i is an access point in the set *APs*.

Let *V* be the set of non-duplicated RSS values measured from ap_i and v_j is each value in the set *V*.

D_{v_j} is the subset of *D*, in which the RSS obtained from ap_i equals v_j and p_m is the probability of a position m obtained from the access point ap_i.

p_m is calculated by dividing the number of samples in subset D_{v_j} which associated to position m by the number of all samples in D_{v_j}.

After information gain of all access points in the set *APs* is obtained, these access points are sorted by their values of information gain. The access points with high information gain illustrate that they are significant to predict the correct positions. These access points are called the prominent access points. In brief, the information gain of the particular AP differ from RSS of the particular AP. The information gain of the AP is used to evaluate effect of this AP to the answer of positioning for all positions. The AP with higher information gain can provide more helpful information, and hence, reduce calculation. Then, the RSS from selected APs is used to determine the specific position. The next step is providing data from the prominent access points to machine learning.

3.2 Clustering model

The clustering model identifies groups of positions divided by similar RSS in that area. This is done without the prior knowledge about the RSS data's characteristics. Such models are often mentioned as unsupervised learning models [15]. There is no external standard to evaluate clustering model's performance. Hence, there are no right or wrong answers for clustering models. Their performance is determined by their ability to merge interesting positions together and to provide descriptions of those groupings. In this work, the K-Means algorithm is selected for clustering phase.

The Figure 1 shows the procedure of K-Means clustering. The K-Means clustering [16] divides positions set into K distinct area or clusters. Firstly, the K number of clustering centers or the centroid points are assigned. Next, the algorithm iteratively assigns data to clusters by the measuring distance from the closest centroid points, and adjusts the centroid points by comparing the distances from each data until further refinement can no longer give the improvement. The K-Means algorithm uses a process known as unsupervised learning [13] to discover patterns in the set of input data. In this work, the clustering model is used for dividing the partitioned areas in order to approximate the rough position of the object of which RSSs are related to the partitioned area.

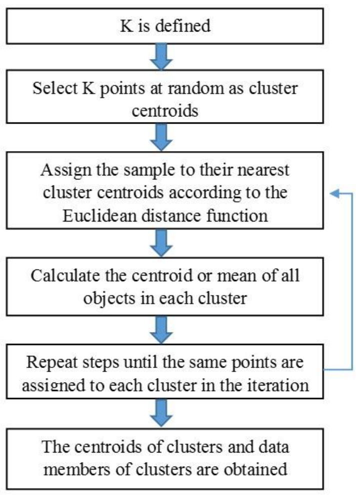

Figure 1 Flowchart of K-Means

3.3 Classifying model

The classification is the problem of identifying to which of a set of positions a new RSS data belongs, based on the observed RSS data and partitioned area data whose positions are known. The classifying model provides the values of positions under prediction, inferred from the value of the class positions [15]. In this work, class variables are the positions in the specific area and the classifying model is the tool to indicate position from the RSS data. The classifying models such as Decision Tree, Artificial Neural Network and Naive Bayes are performed by using WEKA [17], which is the open source software. Their brief details are as follows.

Decision Tree (DT) is a classification algorithm which maps observation data to conclusions about that data's target value or output with these trees' structures [14]. This algorithm, data is split into two or more sets based on the information of gain input attributes. Decision Tree has illustrating ability; e.g. humans can understand the procedure of decision to obtain the output. It requires less data cleaning because it can handle with null value, and it is not influenced by outliers. However, over fitting is one of the most practical problems for Decision Tree.

Naive Bayes (NB) is a probabilistic classifier based on Bayes' theorem with attribute's independence assumed by classifying the class as the one that maximizes the subsequent probability [9]. The main task is estimation the joint probability density function for each class. Naive Bayes is less complex classifier. When attributes are independence, Naive Bayes performs the decent performance. However, in real world problem, it is almost impossible to find completely independent attributes in dataset.

Artificial Neural Network (ANN) is a mathematic model that is represent complex input/output relationships by using learning method similar to a human brain [11]. There are the input layer, hidden layer, and output layer. The hidden nodes are in the hidden layer. The pattern of classification is learned from the training data. The hidden nodes are adjusted to catch that the pattern. Artificial Neural Network is slow algorithms due to large number of hidden nodes. In this work, the multi-layer perceptron neural network is used in the experiment.

Each of algorithms is a component of the proposed Filter Partitioning Machine Learning Classifier. Both of clustering algorithms (K-Means) and classification algorithms (DT, NB, and ANN) perform their task in the process of positioning.

3.4 Filter partition machine learning classifier (FPMLC)

The purposed FPMLC method consists of the feature selection method combined with cascading two components of the machine learning models. The procedure is illustrated in Figure 2. Feature selection is a method that is used to filter informative APs. Then, RSSs of informative APs are fetched to the cascaded model for classification. Positions are classified by two cascading machine learning models. First, the clustering machine learning model divides partitioned areas by using characteristics of RSS data. The partitioned area data can increase information in order to find the position. After partitioned areas or cluster groups are obtained, the classifier determines a position by utilizing the RSS data and the partitioned area.

4. Experiment

In order to evaluate the performance of FPMLC, the experiment is set up in the 30x10 m^2-sized area with the ceiling height of 2.8 m as illustrated in Figure 3. The distances between measured points, i.e. mark points, are one meter grids with 69 reference points (69 classes for classification) from 33 APs. There are 3 APs on this floor and the others are not on this floor. The RSS data was measure by laptop computer, LENOVO Y550/P8800, with the wireless Intel 5100 agn Wi-Fi module. For each reference point, RSS is measured 20 times with 2-second delay. The measuring process is repeated 11 times to create 11 datasets. Hence, this created 69*20*11 = 15,180 samples for measured data.

For standalone models, all 33 APs are included in calculation. For FPMLC, all APs are filtered. After that, top three informative APs are selected for clustering. For comparing with standalone models, i.e. Decision Tree, Artificial Neural Network, and Naive Bayes, three APs that give the strongest RSSs are used in calculation. The experiment is done by using JAVA and the machine learning library from weka software to determine performance before the implementation.

In order to obtain the best performance of FPMLC, data from all APs have to be filtered to discover prominent APs before performing in clustering and classification process. Filtering is done by calculating information gain from APs and sorting order of them from low level to high level. The number of top informative APs which provide more accuracy whereas require a minimum number of APs, are selected for clustering.

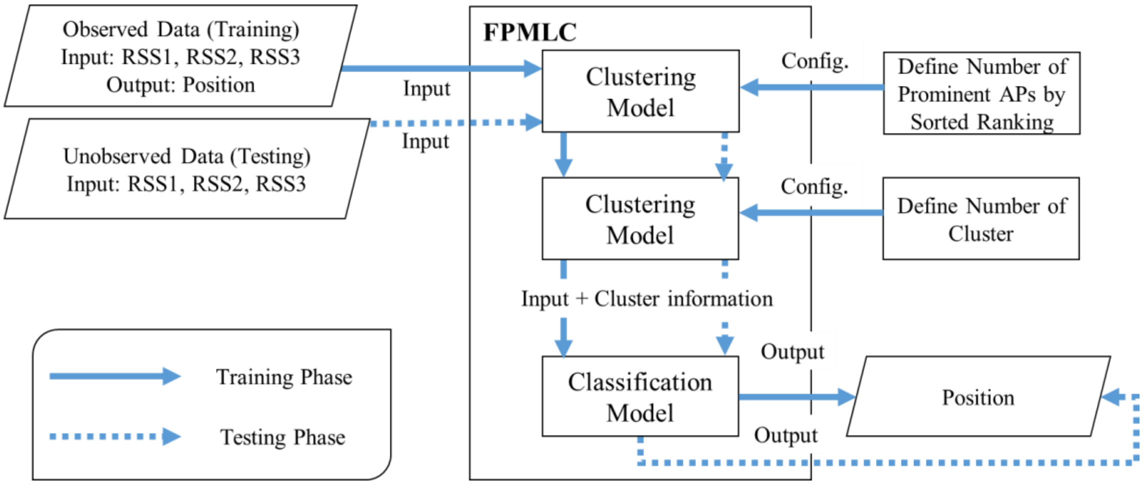

Figure 2 Procedure of Filter Partitioning Machine Learning Classifier

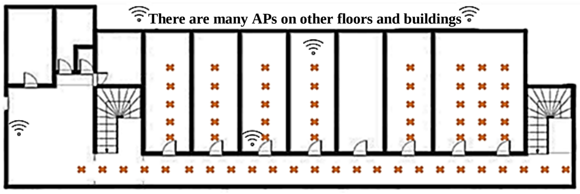

Figure 3 The experimental area

In a clustering process, the appropriate number of clusters has to be discovered. In this work, the numbers of clusters vary from 5 to 10. Then, information of RSSs and clusters are sent to the classification model to figure out the position.

In the classifying process, there are three machine learning algorithms to be employed. Before each classification algorithm is employed, the configuration parameters have to be tuned. In Decision Tree and Naive Bayes, no parameter configuration is necessary. On the other hand, Artificial Neural Network with 4 hidden layers is used. The numbers of hidden node on each hidden layer are 67, 33, 39, and 102 respectively. It must be tuned to find the appropriate parameters. In this experiment, the multi-layer perceptron ANN which has hidden layers is used with the configuration of the learning rate of 0.2, the momentum of 0.2, and the learning cycle of 500 epochs. In addition, every algorithms are trained and test by using 10 fold cross-validation method [18].

After each of the aforementioned classification algorithms is integrated into FPMLC, RSS dataset is used as the training data. Then, FPMLC with different classification algorithms are evaluated by using testing data. Factors of evaluation are used to evaluate the performance. The process of the experiment is shown in Figure 4. The experiment is repeated 10 times to obtain the average result.

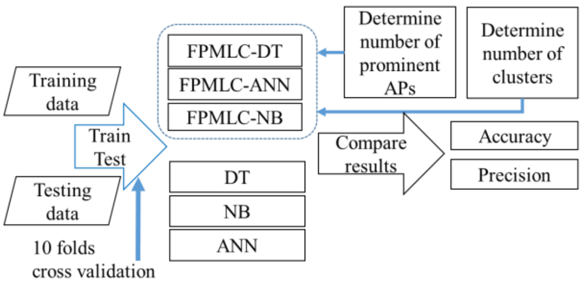

Figure 4 The experimental process

Factors of evaluation are the performance indicators to estimate whether a model is appropriate for positioning. In this work, accuracy and error distance are used as performance indicators.

Accuracy refers to rate of correct positioning. In this work, percent of accuracy is calculated from results of classification that are the identical to reference position. Thus, percent of accuracy is calculated by using equation (2). The number of correct positioning is *NC* and the number of faulty positioning is *NF*.

$$accuracy\% = \frac{NC}{NC+NF} \times 100 \quad (2)$$

Error distance of positioning is evaluated by measuring the Euclidean distance between classified positions and reference positions. A very precise positioning would be less distributed. In this work, error distance is expressed by using standard deviation of error positions and maximum error distances.

In addition, performance in terms of computational complexity is analyzed by using big-O analysis. Cost of each algorithm is calculated by using parameters form the real experiment.

5. Experimental results

In this section, abbreviations are used. Decision Tree is denoted by “DT”. Naive Bayes is denoted by “NB”. Artificial Neural Network is denoted by “ANN”.

The radio map was measured from 33 APs in experimental area. The data contains 1000 samples per measured position. The strongest RSS is -46 dBm and the weakest RSS is -100 dBm. The example of distribution and of RSSs in the study area is shown in Figure 5. The filter in FPMLC filters informative APs by ranking information gain. The first process is discovering an appropriate number of informative APs. The number of the highest information gain AP is varied to obtain the best accuracy of classifying models. Figure 6 illustrates the relation between the number of informative APs and accuracy. We can see from Figure 6 that 5 is the minimum number of informative APs that provides the highest accuracy. This reason leads to less computation in the classification. Therefore, 5 APs are selected in the classification.

Table 1 is the example of information gain of the informative APs. This shows how to arrange APs from information gain. This table shows information gain of the top ten informative APs from all APs. The top five informative APs have explicitly higher information gain than the others. The data from informative APs is selected to cluster in the next process.

After FPMLC obtains the appropriate number selected APs, the number of appropriate clusters needs to be discovered. The number of clusters is varied in the experiment.

The number of clusters obtained from the clustering phase affects positioning accuracy. Table 2 compares percent of accuracy obtained from various numbers of clusters when each of the classifying algorithms is employed in FPMLC. The best accuracy is obtained when 7 clusters are assigned. When 8 clusters are assigned, there is a slightly decline in accuracy. However, accuracy is drastically declined when 5, 6, and 9 clusters are assigned. Hence 7 clusters are the most appropriate number of clusters because it provides the highest accuracy. After that, the cluster information will be provided to the classifier part in FPMLC.

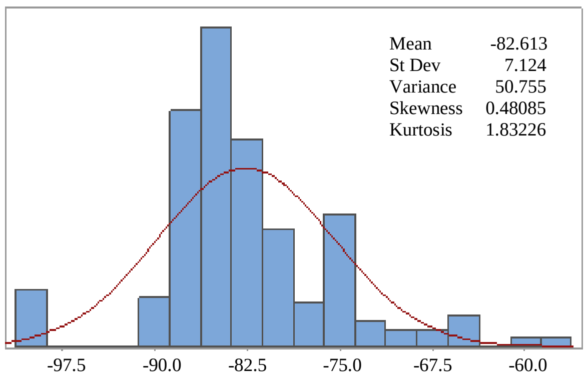

Figure 5 Distribution of RSS in the study area

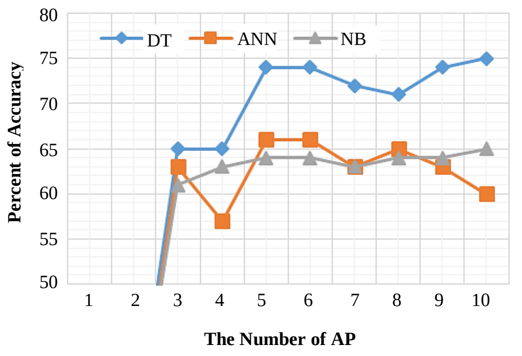

Figure 6 The accuracy compares with the number of informative APs sorted by information gain

Table 1 Samples of the top information gain from the data

The Order of Information Gain	The Value of Information Gain
1st	0.360
2nd	0.295
3rd	0.278
4th	0.267
5th	0.257
6th	0.095
7th	0.077
8th	0.069
9th	0.057
10th	0.051

Table 2 Percent of accuracy obtained from different numbers of clusters

Accuracy [%]	5 clus.	6 clus.	7 clus.	8 clus.	9 clus.
FPMLC-DT	67.5	72.6	78.5	77.3	73.3
FPMLC-ANN	42.2	58.8	72.1	71.8	63.3
FPMLC-NB	47.4	66.8	73.6	72.1	58.3

Table 3 Percent accuracy of each classification algorithm, averaged from 10 repeated experiments

Algorithms	Accuracy [%]
DT	73.61
ANN	65.72
NB	63.61
FPMLC-DT	78.52
FPMLC-ANN	72.1
FPMLC-NB	73.64

Table 4 Standard deviations and maximum error distances

Algorithms	StdDev.	Max.Error [m]
DT	1.137	5
ANN	1.5529	8
NB	1.2884	5
FPMLC -DT	1.0315	3
FPMLC -ANN	1.337	5
FPMLC -NB	1.236	5

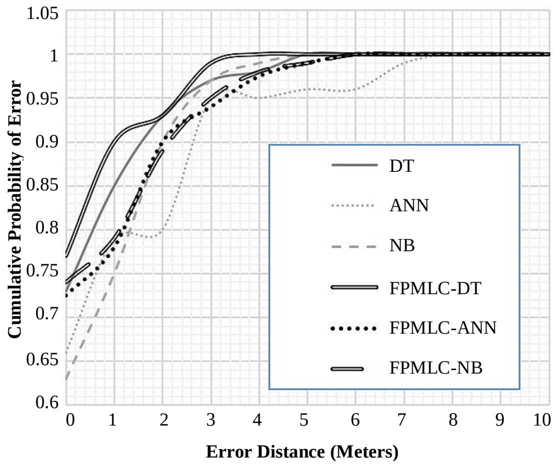

Figure 7 CDF of error distance

Table 3 shows percent of accuracy of FPMLC with different classification algorithms. In comparison with individual classification algorithms, FPMLC can increase accuracy of every classification algorithm around 5 to 10 percent. When FPMLC is built with DT, the accuracy is improved around 4.91 percent compared with the standalone DT. In addition, FPMLC with ANN and FPMLC with NB can increase accuracy around 6.38 percent and 10.03 percent respectively compared with their standalone counterparts.

In terms of error distance, standard deviations (StdDev.) and maximum error distance (Max. Error) are used to consider the error. These values are averaged from 10 repeated experiments. From accuracy evaluation, ANN and NB give almost similar accuracy. However, their standard deviations and maximum error distances are significantly different, as shown in Table 4. Furthermore, it shows that FPMLC can reduce the standard deviations and the maximum error distances of DT and ANN. For FPMLC-NB case, the proposed algorithms can provide useful information that makes learning mechanic of NB improved. However StdDev. and Max.Error of FPMLC-NB is not much better than NB compared with the other FPMLCs. The MAX.Errors of them are the same. The StdDev. of FPMLC-NB is slightly improved from NB. There is some improvement of FPMLC-NB that is discussed in the error distance result.

Figure 7 shows the CDF of error distance. The result shows performances which agree with accuracy and error distance that are mentioned earlier. The algorithms with the higher value of CDF and the smaller error distance would be preferred because it is more possible that small error distance will be obtained. Before FPMLC is applied, all of the standalone algorithms show the value of CDF around 0.9 within 2 meters of error distance except ANN that shows the value of CDF around 0.8 within 2 meters of the error distance. In addition, CDF of ANN reports that the ANN algorithm reaches almost 100 percent probability within 8 meters of the error distance while the others are around 5 meters. After FMLC is applied, FPMLCs can improve positioning performance of their standalone counterparts. Their probability of error distance are higher compared to those of the standalone algorithms. All of FPMLC algorithms show performance with 90 percent probability within 2 meters of the error distance except that FPMLC-DT shows probability around 93 percent. CDF of FPMLC algorithms reports that the probability of FPMLC algorithms reaches almost 100 percent within 5 meters of the error distance. Moreover, FPMLC-ANN is better than standalone ANN around 2 meters. For FPMLC-NB, CDF within 2 meters error distance is better than that of NB. It corresponds to the accuracy of FPMLC-NB, which is improved from NB. However, the FPMLC-NB's probability is slightly lower than NB when the error distance is around 2 to 5 meters.

The computation complexity of FPMLC and the other algorithms are compared as shown in Table 5.

In offline phase, each algorithm is trained by the data. Let N be the number of samples and F be the number of APs. $\hat{\boldsymbol{F}}$is the number of APs after filtering. For K-Means, K is the number of clusters. For ANN, I is the number of calculating iterations and M is the number of hidden nodes. The number of hidden nodes is calculated from the total number of hidden nodes in architecture.

Table 5 Big-O notation and computation cost

algorithm	Big-O notation	cost
K-Means	$O(K*N)$	106260
Filtering	$O(N*F)$	500940
DT	$O(N*F^2)$	16531020
ANN	$O(N*F*M*I)$	60363270000
NB	$O(N*F)$	500940
FPMLC-DT	$O(k*N)+O(N*F)+O(N*\hat{F}^2)$	986700
FPMLC-ANN	$O(k*N)+O(N*F)+O(N*\hat{F}*M*I)$	9145950000
FPMLC-NB	$O(k*N)+O(N*F)+O(N*\hat{F})$	683100

Form Table 5, in normal condition, the computational cost of ANN is the worst because of the effect from a large number of hidden nodes. The NB's computational cost is always less than that of Decision Tree. In fact, if the number of APs are very high, the computation cost of Decision Tree is drastically increased. Therefore, the prominent AP selection in the proposed method helps reducing is the number of APs, and hence, helps reducing computational complexity. However, the complexity of proposed method includes computation of K-Means and AP filtering. In table 5, parameters from real experiment are applied (N = 15,180, F = 33, $\hat{\boldsymbol{F}}$ = 5, K = 7, I = 500, M = 241). Since Figure 6 illustrates that 5 is the minimum number of informative APs that provides the highest accuracy. $\hat{\boldsymbol{F}}$ is 5. By reducing the number of APs, computational cost is reduced. From Table 5, costs of the proposed FPMLCs are lower than those of the standalone counterparts. Even though, the computational cost of FPMLC-NB is a little higher than that of NB, the accuracy of FPMLC-NB is much higher as shown in Table 3.

In online phrase, the positions are predicted from the trained algorithm. The RSS data is scanned from mobile node. Then the sample is classified, where the position of the mobile node should be. This computation of every algorithm is $O(1)$.

In practical, the finger printing positioning is used in the online phase. The energy consumption of this classified process is very low as its big-O is $O(1)$. The energy of the mobile node is used for Wi-Fi scanning to collect sample data. Then, mobile node, which obtained the positioning algorithm, uses that sample for the position prediction.

Overall, the experiment can illustrate that FPMLCs can improve performance for indoor positioning compared with their standalone counterparts. The improvement comes from the information partitioning and the prominent APs selection. The algorithms with better information sources can provide better positioning results.

6. Conclusion

This paper proposes the Filter Partitioning Machine Learning Classifier algorithms for indoor location positioning. FPMLC consists of 3 phases. The first phase is choosing prominent AP by filtering. The second phase is the clustering phase, which employs the K-Means algorithm in order to obtain the appropriate number of clusters. The last phase is the classification phase. Three difference classification algorithms, i.e. Decision Tree, Naive Bayes, and Artificial Neural Network, are used in the comparison. The real data set from an experimental site is used in the performance evaluation. The experimental results show that FPMLC improves each individual classifier in terms of accuracy and error distance. In addition, FPMLC shows the best performance when Decision Tree is employed as the classifier.

Our future work is aimed to improve the algorithm and extend the area of experiment including the multi-floor condition. Due to multi-floor, the RSS is more fluctuated. Hence, the algorithm will be more complex.

7. Acknowledgment

This research is financially supported by Khon Kaen University under the Incubation Research Project.

8. References

[1] Liu H, Darabi H, Banerjee P. Survey of wireless indoor positioning techniques and systems. Systems. IEEE Trans Syst Man Cybern C Appl Rev. 2007;37(6):1067-80.

[2] Lin TN, Lin PC. Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks. 2005 International Conference on Wireless Networks, Communications and Mobile Computing; 2005 Jun 13-16; Maui, USA. USA: IEEE; 2005. p. 1569-74.

[3] Mok E, Retscher G. Location determination using wifi fingerprinting versus wi-fi trilateration. Journal of Location Based Services. 2007;1(2):145-59.

[4] Mautz R. Overview of current indoor positioning systems. Geodesy and Cartography. 2009;35(1):18-22.

[5] Bahl P, Padmanabhan VN. RADAR: An in-building RF-based user location and tracking system. INFOCOM 2000. Proceedings of IEEE 19th Annual Joint Conference of the IEEE Computer and Communications Societies; 2000 Mar 26-30; Tel Aviv, Israel. USA: IEEE; 2000. p. 775-84.

[6] LaMarca A, Chawathe Y, Consolvo S. Place lab: device positioning using radio beacons in the wild. In: Hans WG, Roy Want, Albrecht Schmidt, editors. Pervasive computing. Heidelberg: Springer Berlin Heidelberg; 2005. p. 116-33.

[7] Madigan D, Einahrawy E, Martin RP. Bayesian indoor positioning systems. INFOCOM 2005. Proceedings of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies; 2005 Mar 13-17; Miami, USA. USA: IEEE; 2005. p. 1217-27.

[8] Badawy OM, Hasan MAB. Decision tree approach to estimate user location in WLAN based on location fingerprinting. Proceeding of Radio Science Conference 2007; 2007 Mar 13-15; Cairo, Egypt. USA: IEEE; 2007. p. 1-10.

[9] Brunato M, Battit R. Statistical learning theory for location fingerprinting in wireless LANs. Comput

Network. 2005,47(6):825-45.
[10] Chen RC, Lin YC, Lin YS. Indoor position location based on cascade correlation networks. Proceeding of IEEE International Conference Systems, Man, and Cybernetics (SMC); 2011 Oct 9-12; Anchorage, Alaska. USA: IEEE; 2011. p. 2295-2300.
[11] Borenović MN , Nešković AM. Positioning in WLAN environment by use of artificial neural networks and space partitioning. Ann Telecommun. 2009;64:665-76.
[12] Chen Y, Yang J, Yin J, Chai X. Power-efficient access-point selection for indoor location estimation. IEEE Trans Knowl Data Eng. 2006;18(7):877-88.
[13] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3:1157-82.
[14] Quinlan JR. Induction of decision trees. Mach Learn. 1986;1(1):81-106.
[15] Witten IH, E Frank E, Hall MA. Data mining: practical machine learning tools and techniques. Massachusetts: Morgan Kaufmann; 2011.
[16] Arthur D, Vassilvitskii S. K-means++: the advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics; 2007 Jan 7-9; New Orleans, USA. USA: Society for Industrial and Applied Mathematics; 2007. p. 1027-35.
[17] Holmes G, Donkin A, Witten IH. Weka: a machine learning workbench. Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems; 1994 Nov 29-Dec 2; Brisbane, Australia. USA: IEEE; 1994. p. 357-61.
[18] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence; 1995 August 20-25; Montreal, Canada. USA: Morgan Kaufmann Publishers; 1995. p. 1137-45.