Localization of mobile nodes using Artificial Neural Network

Usha K
M.tech 2nd Year
The Oxford College Of Engineering
Bangalore,Karnataka,India
ushakrishna05@gmail.com

A Chrispin JiJi
Assistant Professor
The Oxford College Of Engineering
Bangalore,Karnataka,India
chrispinjij@gmail.com

Abstract : Provision of accurate and reliable location estimates is the key issue for the proliferation of indoor location oriented services and applications. Hence, localization techniques have been intensively studied in indoor WLAN environment. Our localization method is based on Artificial Neural Network (ANN) which provides high accuracy and robust performance, but has a limitation of slow convergence, high complexity and large memory storage requirement. Hence we introduce Affinity Propagation (AP) clustering algorithm which reduce computation cost and memory overhead, and then explore a properties of radio basis function (RBF) neural network which provide accuracy in localization system. Furthermore these algorithms achieve substantial improvement over other methods. However, the suboptimal convergence problem of clustering ANN algorithm reduces the location accuracy of this method. In this paper we will compare with other ANN models to further improve in localization performance.

Keywords – WLAN, AP clustering, ANN.

I INTRODUCTION

The increasing demand of location based service has promoted the development of indoor positioning systems. Application of indoor positioning can be found widely from medicine and industry to our daily life such as visitor navigation. Currently, the mainstream of wireless technologies adopted in indoor situations includes WLAN, RFID, UWB as well as Bluetooth [1]. However, due to the complex deployment of indoor situations and the presence of non-line-of sight paths, it is not easy to establish an accurate propagation model. Hence, the conventional multilateration or multiangulation techniques may result in larger location error than scene analysis algorithm which is also called fingerprinting localization [1].

In general, fingerprinting localization systems fall into two phases, namely offline phase and online phase. In the offline phase, mobile terminals collect the set of RSS values, which is called the fingerprint, from various access points for each location, and then save these in the database (i. e., a map). During the online phase, the locations of terminals are obtained by matching observations of RSS values collected in real-time to the map of the previous fingerprints. Various matching algorithms can be found in the literature, such as the K-Nearest- Neighbor (KNN) scheme [2], Kernel based algorithm [3], Support Vector Machines (SVM) [4], and Artificial Neural Networks (ANN) based method [5, 6].

As an important alternative, ANN based fingerprinting localization method could potentially achieve higher accuracy performance than other traditional methods [5]. However, when the networks size gets larger, especially in a large-scale fingerprinting localization system, the training of ANN model has the limitation of poor convergence rates and is easy to fall in local optima, which would result in larger location error, and increase the computational complexity. Laoudias et al. [5] proposed clustered radio basis function (cRBF) neural network architecture to locate the mobile terminals. Their analysis and experimental results show that using clustering method could efficiently reduce computational complexity and memory requirements. In general, as K-means clustering method is easy to implement with low time complexity, it is more widely used in the existing localization algorithms [7, 8]. However, it is an iteration process which starts with an initial guess, which is in the condition of close to the true solution to avoid local minima. Selection of such a starting point is not simple in practice [9]. Furthermore, it would affect the location accuracy performance. Affinity Propagation (AP) clustering approach proposed by Frey [10] can overcome such drawbacks. It can automatically select the number of clusters, clustering by message passing iteratively between fingerprints, and eventually generate a good set of exemplars [10].

According to this theory, in this work, we proposed an AP based clustering method to partition the fingerprint database collected in the offline phase, and then adopt the RBF networks to train the ANN models for each clusters. In the online phase, to improve the real-time capability and reduce the overhead of the localization system, we estimate the mobile terminals' locations by using cluster matching scheme firstly and then using ANN to achieve the final location coordinates. The experiments, performed in a real WLAN environment, demonstrate that clustering method has great influence upon the performance in terms of location accuracy, computation complexity and memory overhead. The results also show that the proposed AP clustering based ANN (APCANN) algorithm reduces the offline training time of the ANN models, and outperforms other methods.

The rest of this paper is organized as follows. Section II describes the localization system structure. Section III introduces the offline clustering by affinity propagation approach and the training model using RBF networks in detail. Experimental setup is present in Section IV. Finally, the concluding remarks and future work are provided in Section V.

II. OVERVIEW OF PROPOSED LOCALIZATION SYSTEM

Currently, to mitigate the effect of multipath in a harsh indoor environment and to improve the location accuracy, RSS-based fingerprinting localization algorithms have been extensively studied in the literature. The basic fingerprinting architecture consists of offline training stage and online location determination stage.

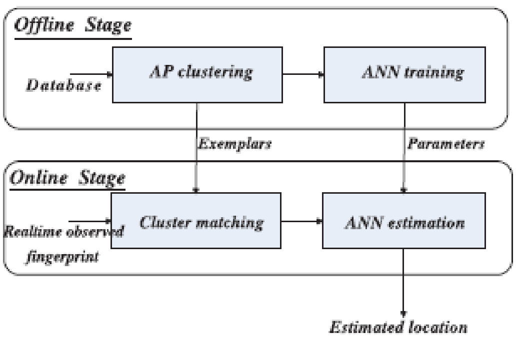


Fig. 1. Proposed indoor localization system

Fig.1 shows the overview of the proposed localization system. The offline stage performs two tasks: clustering and ANN training. Firstly, a set of RSS values are collected at the predefined reference point (RP_i) $l_i = (x_i, y_i)$, i = 1, 2, ...,L from N available access points, and L is the total number of RPs. A series of RSS values collected at l_i are denoted as $v_i(\tau) = [v_{i,1}(\tau), ...v_{i,N}(\tau)]^T$, τ = 1, ..., t, t >1, where $v_{i,j}(\tau)$ denotes the RSS value collected at time τ from the j-th access point, t is the sampling period, and N is the number of detected access points. Then all the RSS time samples are stored in a database. In addition, the average of the series RSS samples is computed and makes a radio map which is defined as V:

$$V = [\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_L] = \begin{bmatrix} v_{1,1} & v_{2,1} & \cdots & v_{L,1} \\ v_{1,2} & v_{2,2} & \cdots & v_{L,2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{1,N} & v_{2,N} & \cdots & v_{L,N} \end{bmatrix}$$

where $v_{i,j} = \frac{1}{t}\sum_{j=1}^{t} v_{i,j}(\tau)$.

In order to reduce the overhead and the computing complexity of the system, AP clustering algorithm then works on the V and partitions the RPs into different clusters represented by an exemplar. During the process of clustering, all RPs are equally considered to be potential exemplars, and then affinity propagation transmits measured similarity messages recursively between pairs of RPs until exemplars and corresponding clusters are generated. For each cluster, we define C_k, k= 1, 2, ...,K, as the set of the k^{th} cluster's members, and c_k as the exemplar, respectively, where K is the total number of the clusters.

When the clustering is completed, the training of artificial neural network is carried out for each cluster. The ANN includes an input layer, corresponding to the RSS values collected from the N available access points, a single hidden layer with non-linear radial basis functions and an output layer representing the two dimensional location (x, y) of the mobile terminal. The main idea of the ANN is to minimize an error function to adjust the internal weights of the network as shown in Section III.

The entire optimized network parameter, including the structures of the networks and the weights vector of the corresponding RPs, then would be stored in the database with reference RP location for the online stage.

In the online localization stage, the unknown location ˜l is obtained by two phases. First, we operate a pattern matching to compute the similarity between the new collected RSS signal

Fig. 1. Proposed indoor localization system vector ˜v˜l= [˜$v_{\tilde{l},1}$, ˜$v_{\tilde{l},2}$, ..., ˜$v_{\tilde{l},N}$]^T and the cluster exemplars stored in the database. The similarity function can be defined to be the squared Euclidean distance, that is

$$s = -\|\tilde{\mathbf{v}}_{\tilde{l}} - \mathbf{c}_k\|^2, k = 1, 2, ..., K$$

Then we choose β(β _ 1) sets of best-matched clusters for the next ANN localization phases to avoid the edge problem[11], which would result in large bias estimation when the mobile terminal is at the boundaries of clusters. What is more, the real-time collected fingerprint ˜l is carried out as the input of previous trained ANN networks for each $RP_i(l_i)$ in the selected clusters, with the corresponding output location estimation ˆl_i (ˆx_i, ˆy_i).

Subsequently, we choose m different estimated locations with the least estimation error
$d(l_i, \hat{l}_i) = \|l_i - \hat{l}_i\|^2$, and then calculate the estimated positioning formation $\tilde{l}$ as follows

$$\tilde{l} = \frac{1}{\sum_{i=1}^{m} \alpha_i} \sum_{i=1}^{m} \alpha_i \hat{l}_i$$

$$\text{where } \alpha_i = \frac{1}{\delta + d(l_i, \hat{l}_i)},$$

and δ is a small real constant ($\delta = 0.01$
in our tests) used to avoid division by zero.

III. AFFINITY PROPAGATION CLUSTERING BASEDNEURAL NETWORKS

In addition to location accuracy, low complexity and memory overhead are also the key requirements for a ubiquitous sand practicable application of ANN based fingerprinting localization system. Laoudias [5] reported that clustering scheme can effectively reduce the size and computational complexity of the ANN system. However, the selection of clustering algorithm affects the location performance seriously. In the following subsection, we present an affinity propagation clustering algorithm which results in a better clustering performance and obtains higher location accuracy as shown in the following
section IV.

A. Affinity propagation clustering

Affinity propagation, proposed by Frey and Dueck [10], performs well in many applications such as gene expression, image categorization, and document clustering. Instead of randomly selecting K initial exemplars in the K-means clustering algorithm, AP initially considers all RP as potential cluster exemplars, and then exchanges messages between RPs until a stable state is reached. Clusters are formed by assigning each RP to its most similar exemplar.

The input of affinity propagation is the similarity matrix $S$and preference which is denoted as P. Matrix S consists of all the pairwise similarity $s(i, j)$ and self-similarity $s(i, i)$, and the S in this paper is defined as

$$s(i,j) = -\|\mathbf{v}_i - \mathbf{v}_j\|^2$$

$$s(i,i) = P = \gamma \cdot median\{s(i,j)\}$$
)

Where $\forall i, j \in 1, 2, ..., L$, $i\_= j$, and the preference P is a fixed value which is set γ times the median of the pairwise similarities.

The process of AP, as shown in Table I, can be viewed as an exchanging process of two kinds of messages, named responsibility and availability, respectively. The responsibility $r(i, j)$, sent from RP_i to potential exemplar RP_j, reflects
how suitably RP_j serve as the exemplar for RPi. The availability $a(i, j)$, sent from potential exemplar RP_j to RP_i, reflects how willingly RP_i admits RP_j as its exemplar. The message exchanging procedure may be terminated when the
number of iterations reaches a predefined value denoted as *num_iterations*, or the decision results stay constant for a fixed number of iterations [10]. The method also introduces a damping factor λ to avoid numerical oscillations during the procedure of updating messages. Throughout the following of
this paper, the damping factor is set a default value λ = 0.5according to Frey's reports [10].

B. Radial basis function neural networks

In general, a neural network is mainly made up of input, output, and hidden layers. The overall structure, shown inFig.3, implements a nonlinear mapping $F : R^N \rightarrow R^2$expanded on a finite basis of nonlinear functions. Usually, we choose Gaussian functions as the radial basis functions. Then the networks can be expressed in the form as

TABLE I
AFFINITY PROPAGATION CLUSTERING ALGORITHM

Input: Preference parameter γ and Similarity matrix S
Initialization: set $a(i,j) = 0$,$r(i,j) = 0, \forall i,j$;
for k=1:*num_iterations*
    if stopping condition is not met
        $r_{new}(i,j) = S(i,j) - max_{j' \neq j}(a(i,j') + S(i,j'))$;
        $a_{new}(i,j) = min\{0, r(j,j) + \sum_{i' \neq i,j} max(0, r(i',j))\}, i \neq j$;
        $a_{new}(j,j) = \sum_{i' \neq j} max(0, r(i',j))$;
        Responsibility updates:
            $r(i,j) = \lambda \times r(i,j) + (1-\lambda) \times r_{new}(i,j)$
        Availability updates:
            $a(i,j) = \lambda \times a(i,j) + (1-\lambda) \times a_{new}(i,j)$
    end
end
Exemplar identification:For all RP_k with $a(k,k) + r(k,k) > 0$ are the identified exemplars.
Cluster assignments:Assign other RP_i to the closest exemplar under similarity measurement.

$$f(\mathbf{v}(\tau)) = \sum_{i\in\mathbb{C}} \mathbf{w}_i h_i(\mathbf{v}(\tau)) = \sum_{i\in\mathbb{C}} \mathbf{w}_i \frac{\hbar_i(\mathbf{v}(\tau))}{\sum_{j\in\mathbb{C}} \hbar_j(\mathbf{v}(\tau))}$$

$$\text{where } \hbar_i(\mathbf{v}(\tau)) = exp\{-\frac{\|\mathbf{v}(\tau)-\mathbf{u}_i\|^2}{2\sigma_i^2}\}$$

$v(\tau) \in R^N$ is the input vector denoted as $[v1(\tau), ..., v_N(\tau)]T$,τ = 1, 2, ..., t, $f(v(\tau))$is the outputs representing the two dimensional geographic position (x, y), w_i is the synaptic weights, u_i and σ_i are the center and the width, respectively, of the normalized Gaussian basis function $hi(v(\tau))$, and C denotes the cluster generated by AP clustering algorithm.

During the offline training for ANN model, the parameters u_i and σ_i affect the performance of the networks since they determine the accuracy of the fit between the function approximation in Eq.(5) and the reference data [5]. In this paper, we set u_i equal to the mean value fingerprints $u_i = v_i$, and set the width$\sigma_i^2 = d_{max}/\sqrt{2n}$ [12], where i = 1, 2, ...,L,n is the size of the cluster C, and $d_{max} = max \|u_i - u_j\|$
$i, j \in C$, $i\_= j$. However, according to the Fig.4 in [5], when there is only one cluster member in C, the width is set equal to a default value $\sigma_i^2 = 50$ to minimize the mean location error. Then, to obtain the optimal weights w_i, a recursive least squares method or a gradient search algorithm, such as Levenberg-Marquardt algorithm, can be applied to minimize the following objective function

$$F(\mathbf{W}) = \sum_{\tau=1}^{t}(l_i - \sum_{i\in\mathbb{C}} \mathbf{w}_i h_i(\mathbf{v}_i(\tau)))^2$$

where **W** = $[w_i, ..., w_j]^T$, $i, j \in C$. Subsequently, the optimal weights **W**are used in the online phase to estimate the location ˆl given a new fingerprint ˜v according to

$$\hat{l} = \sum_{i\in\mathbb{C}} \mathbf{w}_i h_i(\tilde{\mathbf{v}})$$

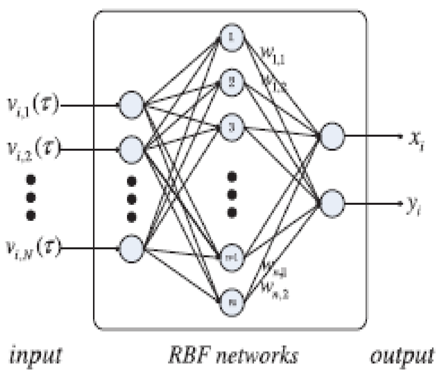


Fig. 3. Structure of artificial neural networks

C. Location accuracy performance analysis

Laoudas et al. [5] analyze the advantages of the clustering ANN fingerprinting localization algorithm in training rate, memory overhead and computation cost. However, a better clustering algorithm could further improve the performance
which is presented in the next section. In this subsection, we will describe the affection of clustering ANN algorithms on the location accuracy. The network mapping from the hidden space to the output space is designed to perform a linear mapping and the linear simultaneous equations for the unknown weights w_i can be defined as:

$$\mathbf{L}_{1\times 2} = \mathbf{H}_{1\times n}\mathbf{W}_{n\times 2} + \varepsilon$$

where H = $[hi(v) \cdot \cdot \cdot hj(v)]$, L =
x y, $i, j \in C$ and ε denotes the error between the actual output and the expected output of the ANN model. Obviously, there are many available solutions of the above equations which means that the minimization of the Eq. (6) will have multiple optimal solutions, and will converge to the adjacent area near the initial values of the W [13]. Furthermore, the random initialization of the weights would result in different performances [12]. For instance, the real-time fingerprint ˜v may minimize the output bias of two different trained RBF networks of different clusters which are geographically far away from each other. According to the final location determination of the proposed algorithm, it would lead to a large location estimation bias. We define this phenomenon as the suboptimal convergence problem of the clustering ANN algorithm.

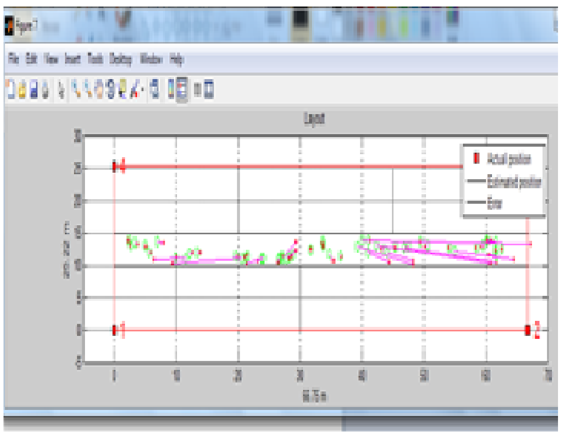


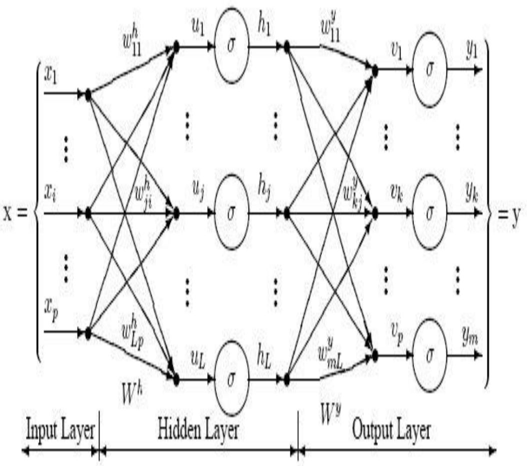
Fig shows the positioning performance of the proposed algorithm

D. Feed forward backpropogation neural network

The first term, "feedforward" describes how this neural network processes and recalls patterns. In a feedforward neural network, neurons are only connected foreword. Each layer of the neural network contains connections to the next layer (for example, from the input to the hidden layer), but there are no connections back. The term "backpropagation" describes how this type of neural network is trained. Backpropagation is a form of supervised training. When using a supervised training method, the network must be provided with both sample inputs and anticipated outputs. The anticipated outputs are compared against the actual outputs for given input. Using the anticipated outputs, the backpropagation training algorithm then takes a calculated error and adjusts the weights of the various layers backwards from the output layer to the input layer.

E. Multilayer Perceptron

The following diagram illustrates a perceptron network with three layers:

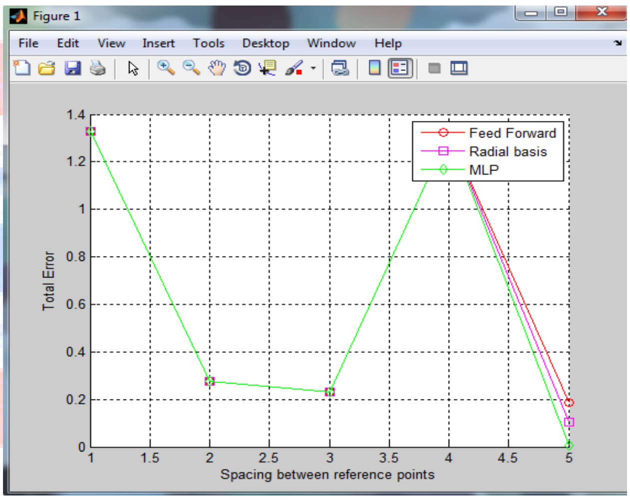


This network has an **input layer** (on the left) with three neurons, one **hidden layer** (in the middle) with three neurons and an **output layer** (on the right) with neurons. There is one neuron in the input layer for each predictor variable. In the case of categorical variables, *N*-1 neurons are used to represent the *N* categories of the variable.

Input Layer — A vector of predictor variable values ($x_1 ... x_p$) is presented to the input layer. The input layer (or processing before the input layer) standardizes these values so that the range of each variable is -1 to 1. The input layer distributes the values to each of the neurons in the hidden layer. In addition to the predictor variables, there is a constant input of 1.0, called the *bias* that is fed to each of the hidden layers; the bias is multiplied by a weight and added to the sum going in to the neuron.

Hidden Layer — Arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight (w_{ji}), and the resulting weighted values are added together producing a combined value u_j. The weighted sum (u_j) is fed into a transfer function, σ, which outputs a value h_j. The outputs from the hidden layer are distributed to the output layer.

Output Layer — Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight (w_{kj}), and the resulting weighted values are added together producing a combined value v_j. The weighted sum (v_j) is fed into a transfer function, σ, which outputs a value y_k. The *y* values are the outputs of the network.



Comparison of different ANN for reduction of location error

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a localization algorithm using affinity propagation clusteringin conjunction with artificial neural networks. Experimentalresults demonstrate that affinity propagation clustering providesan efficient clustering result which makes the training of the ANN model faster and reduces the memory overhead in potential. Furthermore, the proposed algorithm achievessubstantial improvements on localization accuracy over othermethods. However, the suboptimal convergence problem of theclustering ANN algorithm reduces the localization accuracyof the proposed method. Hence, in the future work, we willconcentrate on furtherimprove in localization performance.

VI. REFERENCES

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Surveyof wireless indoor positioning techniques and systems," *IEEE Trans. System, Man, and Cybernetics*, vol. 37, no. 6, pp. 1067–1080, 2007.

[2] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *IEEEINFOCOM*, vol. 2, 2000, pp. 775–784.

[3] A. Kushki, K. N. Plataniotis, and A. N. Eenetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp.689–705, 2007.

[4] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Comput.Netw*, vol. 47, pp. 825–845, 2005.

[5] C. Laoudias, P. Kemppi, and C. Panayiotou, "Localization using radial basis function networks and signalstrength fingerprints in WLAN," in *IEEE GLOBECOM*, 2009, pp. 1–6.

[6] C. Nerguizian, C. Despins, and S. Affes, "Geolocation in mines with an impulse response fingerprinting technique and neural networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 3, pp. 603–611, 2006.

[7] S. Kuo, B. Wu, W. Peng, and Y. Tseng, "Clusterenhancedtechniques for pattern-matching localization systems," in *IEEE MASS*, 2007,.

[8] M. A. Youssef, A. Agrawala, and A. U. Shankar, "Wlanlocation determination via clustering and probability distributions,"
in*IEEE PerCom*, 2003, pp. 143–150.

[9] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans*, vol. 38, no. 1, pp. 218–237, 2008.

[10] B. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976,2007.

[11] C. Feng, W. Au, S. Valaee, and Z. Tan, "Compressive sensing based positioning using RSS of WLAN access points," in *IEEE INFOCOM*, 2010, pp. 1–9.

[12] S. Haykin, *Neural networks - a comprehensive foundation*, 2nd ed. Prentice Hall, 1998.

[13] X. Zhang, *Matrix analysis and application*. Springer and Tsinghua University Press, 2004.

[14] H. Yu, T. Xie, S. Paszczynski, and B.Wilamowski, "Advantages of radial basis function networks for dynamic system design," *IEEE Trans. Ind. Electron.*, vol. 58,
no. 12, pp. 5438–5450, 2011.

[15] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.