2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Constraint Online Sequential Extreme Learning Machine for Lifelong Indoor Localization System

Yang Gu, Junfa Liu, Yiqiang Chen, Xinlong Jiang
Department of Pervasive Computing
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China

***Abstract*—As an important technology in LBS (Location Based Services) field, Wi-Fi based indoor localization suffers signal fluctuation problem which prevents lifelong and high performance running. With the fluctuation of wireless signal over time, fingerprints collected at the same location become different; therefore existing model cannot fit the new collected data well, which decreases the localization accuracy. In this paper, a novel indoor localization method COSELM (Constraint Online Sequential Extreme Learning Machine) is proposed, utilizing incremental data to update the old model and overcome the fluctuation problem. The performance of COSELM is validated in real Wi-Fi indoor environment. Compared with OSELM, it can improve more than 5% localization accuracy on average; and in contrast to batch learning, COSELM can save more than 50% time consumption.**

Keywords—Wi-Fi indoor localization; fluctuation; online learning; lifelong

I. Introduction

Indoor localization based on Wi-Fi draws increasing attention for its easy implementation and numerous practical applications. In general, there are two kinds of indoor localization methods: propagation based method [1] and fingerprint based method [2]. The propagation based method utilizes the nonlinear fading characteristic of Wi-Fi signal to establish Received Signal Strength (RSS) and distance propagation model, which is further used to estimate distances and user's locations. Fingerprint based method usually has two phases: off-line training phase and online locating phase. At training phase, features are extracted according to a certain rule, and the relationship between features and their corresponding locations is established, which constructs a "radio map"; at online locating phase, features of unknown location are extracted with the same rule, and pattern recognition methods are adopted to find the best matched locations on the "radio map" to get the final estimated location. As [3] summarized, "Fingerprint based methods usually produce the most accurate estimation of position in indoor environment", therefore fingerprint methods are widely adopted.

However, there are some practical problems hinder the further prosperity of fingerprint based Wi-Fi localization, and the fluctuation of Wi-Fi signal is a major one. With time passing by, the unpredictable movement of people, the status of door open/closed, the change of temperature and humidity, the placement of obstacles and so on, which all lead to the fluctuation of Wi-Fi signal. [4] specifically shows the effect of dynamic environment that the changing status of door can lead to poor positioning accuracy of 4.59-7.17 meters; the changing status of relative humidity can increase the localization error by almost 20%. [5] investigates the properties of RSS in detail, it shows that people's presence can spread the range of RSS by increasing RSS's deviation. [6] also explains the body effect of human that the distortion caused by human body in different orientations can easily confuse real location with other locations. Works of indoor localization show the severe fluctuation of RSS that the difference between maximum and minimum value of collected RSS at the same spot can be bigger than 10dbm [5,6], which makes existing model perform badly over time.

In order to achieve good performance of localization system for long time, Constraint Online Sequential Extreme Learning Machine (COSELM) is proposed, leveraging incremental data to update existing model, so that the new model can cover more possibilities of fingerprints at the same location and reflect localization environment better. The organization of the rest of paper is: Section II shows the related work. Section III introduces the proposed COSELM in detail. Section IV evaluates the performance of COSELM in real wireless indoor environment. Section V concludes the work.

II. Realted Work

Fingerprint method is the mainstream of Wi-Fi based indoor localization. Most of the traditional fingerprint methods are batch learning methods, such as K Nearest Neighbor (KNN) [7], Decision Tree (DT) [8], Support Vector Machine (SVM) [9] and so on. However, there are some disadvantages of batch learning based fingerprint method. One is that it needs training data fully prepared to learn a model. While, for indoor localization, preparing all the data in advance is laborious and time-consuming; especially when the indoor environment is large, it is impossible to collect all the needed data ahead of time. What is more, when new training data are collected to update the model, all the training data have to be used to recalculate the model, whose time consumption is undesirable for real-time localization systems. Hence, online (incremental) methods are applied to indoor localization field, so that not only the time consumption can be reduced, but also the fluctuation problem can be mitigated. [10] presents a two-fold online localization approach: first setting up an initial propagation model with mutual measurements from each base station, then utilizing online learning technology based on Kohonen self organizing map to gradually refined the model.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

In [11], timeliness management has been brought into online incremental learning; considering the signal fluctuation over time, newly incremental data are given bigger weight than old data in generating new model, so that the updated model can fit the current environment and improve localization performance.

To handle the signal fluctuation problem over time, there are many proposed methods. [12] presents LuMA method based on manifold alignment to learn the mapping between source data and target data (source data and target data are collected at different time) in a low dimensional space, so that the knowledge can be transferred to target data to guarantee the localization performance with time passing by. [13] proposes semi-supervised HMM (Hidden Markov Model) method to transfer the out-of-date model to fit a current model; it first learns the regression weights among RSS data from reference points, then builds a radio map at each non-reference point location at new time with the same regression weights constraint, and finally utilizes expectation-maximization (EM) algorithm to get the new localization model. Some works also use adaptive learning methods to solve the fluctuation problem. [14] re-measures RSS fingerprints at a specific location, then calculates the new RSS fingerprints and new locations with plane-interpolation method; with this adaptive RSS database, the algorithm can achieve 80% of positioning correctness in the simulated environment. [15] puts forwards a method using inter-beacon measurement, so that beacons not only serve as localization tools, but also serve as calibration tools to self-adjust the radio map, and experimental results show that both Clustering-based scheme and Regression-based scheme can overcome the fluctuation problem.

Comparing with the existing works, we propose an easy-implemented online indoor localization method COSELM with fast learning speed and high localization accuracy, which can handle the fluctuation problem, and help the localization system function well for a long time.

III. METHODOLOGY

Huang et al. [16] proposed Extreme Learning Machine (ELM) in 2004 to achieve fast learning speed and high performance; however, it just guarantees the minimum error and lacks of generalization ability. Hence, in 2012, they updated the original ELM with L2 constraint [17], so that both the generalization ability and minimum error can be ensured (here we use ELM-C to represent ELM with L2 constraint). ELM itself is a batch learning method, and its online learning version was proposed by Liang [18], which is Online Sequential Extreme Learning Machine (OSELM); OSELM also just guarantees the minimum error criterion. Considering the generalization ability of OSELM, we propose COSELM (The relationship between COSELM and OSELM is the same as the relationship between ELM-C and ELM).

Assuming the initial data are $X_0=\{(x_i,t_i)\}_{i=0}^{N_0}$, where $x_i=[x_{i1},x_{i2},...,x_{in}]$, $t_i=[t_{i1},t_{i2},\ldots,t_{im}]$, n is the dimension of input vector (feature), m is the dimension of output vector (label), N_0 is the number of initial data. COSELM neural network with L hidden neurons can be illustrated in Fig.1.

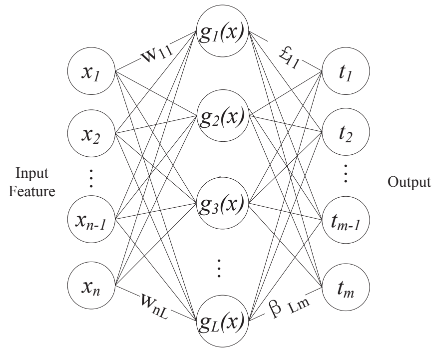


Fig. 1. SLFN with L hidden neurons

The output of this network is as follows:

$$f_L(x)=\sum_{i=1}^{L}\beta_i G(w_i,b_i,x), w_i\in\Re^n, b_i\in\Re, \beta_i\in\Re^m \tag{1}$$

Where $w_i=[w_{i1},w_{i2},...,w_{in}]$ is the weight vector connecting the i^{th} hidden neuron and input neurons, b_i is the bias of the i^{th} hidden neuron, and $G(w_i,b_i,x)$ is the output of the i^{th} hidden neurons. $\beta_i=[\beta_{i1},\beta_{i2},\cdots,\beta_{im}]^T$ is the weight vector connecting the i^{th} hidden neuron and the output neurons.

If the activation function of hidden neurons is $g(x)$, then the output of the i^{th} hidden neuron is:

$$G(w_i,b_i,x)=g(w_i\cdot x+b_i), w_i\in\Re^n, b_i\in\Re \tag{2}$$

The above two equations can be summarized as:

$$H\beta=T \tag{3}$$

Where

$$H=\begin{bmatrix} G(w_1,b_1,x_1) & \cdots & G(w_L,b_L,x_1)\\ \vdots & \ddots & \vdots\\ G(w_1,b_1,x_{N_0}) & \cdots & G(w_L,b_L,x_{N_0})\end{bmatrix}_{N_0\times L}=\begin{bmatrix} h(x_1)\\ \ldots\\ h(x_{N_0})\end{bmatrix},$$
$$\beta=\begin{bmatrix}\beta_1^T\\ \vdots\\ \beta_L^T\end{bmatrix}_{L\times m}, T=\begin{bmatrix} t_1^T\\ \vdots\\ t_{N_0}^T\end{bmatrix}_{N_0\times m} \tag{4}$$

In order to guarantee the generalization ability and minimum error, L2 constraint is brought in, and the target becomes:

$$\begin{aligned}&Minimize: \frac{1}{2}\|\beta\|^2+C\frac{1}{2}\sum_{i=1}^{N_0}\|\xi_i\|^2\\ &subject\ to: h(x_i)\beta=t_i^T-\xi_i^T \quad i=1,......,N_0\end{aligned} \tag{5}$$

Where ξ_i is the training error vector of training data x_i, specifically $\xi_i=[\xi_{i1},\xi_{i2},...,\xi_{im}]$. As shown in [17], based on KKT theorem, the equivalent dual optimization problem is:

$$L=\frac{1}{2}\|\beta\|^2+C\frac{1}{2}\sum_{i=1}^{N_0}\|\xi_i\|^2-\sum_{i=1}^{N_0}\sum_{j=1}^{m}\alpha_{ij}\left(h(x_i)\beta_j-t_{ij}+\xi_{ij}\right) \tag{6}$$

There are three parameters involved in (6): the Lagrange multiplier $\alpha=\left[\alpha_1,\alpha_2,...,\alpha_{N_0}\right]$ and $\alpha_i=\left[\alpha_{i1},\alpha_{i2},...,\alpha_{im}\right]$; the output weight matrix β; and the training error ξ_i. Therefore the KKT corresponding optimality conditions become:

$$\frac{\partial L}{\partial \beta_j}=0 \to \beta=H^T\alpha \tag{7a}$$

$$\frac{\partial L}{\partial \xi_i}=0 \to \alpha_i=C\xi_i \tag{7b}$$

$$\frac{\partial L}{\partial \alpha_i}=0 \to h(x_i)\beta-t_i^T+\xi_i^T=0 \tag{7c}$$

By submitting the solutions of 7(a) and 7(b) into 7(c), we obtain the output weight, which is:

1) $\beta=H^T\left(\frac{I}{C}+HH^T\right)^{-1}T$, when the number of training samples is not huge (HH^T is invertible).

2) $\beta=\left(\frac{I}{C}+H^TH\right)^{-1}H^TT$, when the number of training sample is huge (H^TH is invertible).

Therefore, for initial data, the output weight becomes:

$$\beta^0=\left(\frac{I}{C}+H_0^TH_0\right)^{-1}H_0^TT_0 \tag{8}$$

Since solution 1) has the same deduction as solution 2), here we just take the solution 2) for example. Making $K_0=\frac{I}{C}+H_0^TH_0$, then :

$$\beta^0=K_0^{-1}H_0^TT_0 \tag{9}$$

Given another chunk of data $X_1=\left\{(x_i,t_i)\right\}_{i=N_0+1}^{N_0+N_1}$, N_1 is the number of data; with all the training data the target becomes :

$$\begin{aligned}&Minimize: \frac{1}{2}\left\|\beta^1\right\|^2+C\frac{1}{2}\sum_{i=1}^{N_0+N_1}\left\|\xi_i\right\|^2\\&subject\ to: \begin{pmatrix}H_0\\H_1\end{pmatrix}\beta^1=\begin{pmatrix}T_0\\T_1\end{pmatrix}-\begin{pmatrix}\xi^0\\\xi^1\end{pmatrix}\end{aligned} \tag{10}$$

Where H_1 is the result when X_1 goes through the network, T_1 is the output vector of these N_1 incremental data, and ξ^1 is the training error for X_1. Utilizing both X_0 and X_1, we get the new output weight, which is $\beta^1=K_1^{-1}\begin{pmatrix}H_0\\H_1\end{pmatrix}^T\begin{pmatrix}T_0\\T_1\end{pmatrix}$. This calculation procedure is the similar to the obtaining of β^0 in (9), except for one major difference that K_1 here is $K_1=\frac{I}{C}+\begin{pmatrix}H_0\\H_1\end{pmatrix}^T\begin{pmatrix}H_0\\H_1\end{pmatrix}=\frac{I}{C}+H_0^TH_0+H_1^TH_1=K_0+H_1^TH_1$. In order to find the relationship between β^1 and β^0, the item of $\begin{pmatrix}H_0\\H_1\end{pmatrix}^T\begin{pmatrix}T_0\\T_1\end{pmatrix}$ is rewritten as:

$$\begin{aligned}\begin{pmatrix}H_0\\H_1\end{pmatrix}^T\begin{pmatrix}T_0\\T_1\end{pmatrix}&=H_0^TT_0+H_1^TT_1=K_0K_0^{-1}H_0^TT_0+H_1^TT_1\\&=K_0\beta^0+H_1^TT_1\\&=\left(K_1-H_1^TH_1\right)\beta^0+H_1^TT_1\\&=K_1\beta^0-H_1^TH_1\beta^0+H_1^TT_1\end{aligned} \tag{11}$$

With the expression above, β^1 becomes:

$$\begin{aligned}\beta^1&=K_1^{-1}\begin{pmatrix}H_0\\H_1\end{pmatrix}^T\begin{pmatrix}T_0\\T_1\end{pmatrix}\\&=K_1^{-1}\left(K_1\beta^0-H_1^TH_1\beta^0+H_1^TT_1\right)\\&=\beta^0+K_1^{-1}H_1^T\left(T_1-H_1\beta^0\right)\end{aligned} \tag{12}$$

From (12), we can see that the effect of incremental data on output weight matrix is reflected by $K_1^{-1}H_1^T\left(T_1-H_1\beta^0\right)$. By this analogy, when the $(k+1)^{th}$ chunk of data arrive, where $X_{k+1}=\left\{(x_i,t_i)\right\}_{i=\left(\sum_{j=0}^{k}N_j\right)+1}^{\sum_{j=0}^{k+1}N_j}$, the output weight becomes:

$$\begin{aligned}&\beta^{k+1}=\beta^k+K_{k+1}^{-1}H_{k+1}^T\left(T_{k+1}-H_{k+1}\beta^k\right)\\&K_{k+1}=K_k+H_{k+1}^TH_{k+1},\ K_0=\frac{I}{C}+H_0^TH_0\end{aligned} \tag{13}$$

To sum up, if the existing model is expressed as β, and the new model is β^*, then the relationship between β^* and β is $\beta^*=\beta+\Delta\beta$, where $\Delta\beta$ is the modification caused by newly incremental data.

According to [18], the goal of OSELM is

$$Minimize: \left\|H\beta-T\right\| \tag{14}$$

After $k+1$ times incremental learning, the solution of OSELM is expressed as:

$$\begin{aligned}&\beta^{k+1}=\beta^k+K_{k+1}^{-1}H_{k+1}^T\left(T_{k+1}-H_{k+1}\beta^k\right)\\&K_{k+1}=K_k+H_{k+1}^TH_{k+1},\quad K_0=H_0^TH_0\end{aligned} \tag{15}$$

Compared the result of COSELM with OSELM, the L2 regularization leads to the item $\frac{I}{C}$. When C approximates ∞, $\frac{I}{C}+H_0^TH_0\approx H_0^TH_0$. Therefore OSELM is a special case of COSELM. Since the L2 constraint makes the model more generalized, and the value of C can be searched from range $\left[-\infty,\infty\right]$ to get the optimized result in (5), COSELM should have better performance than OSELM.

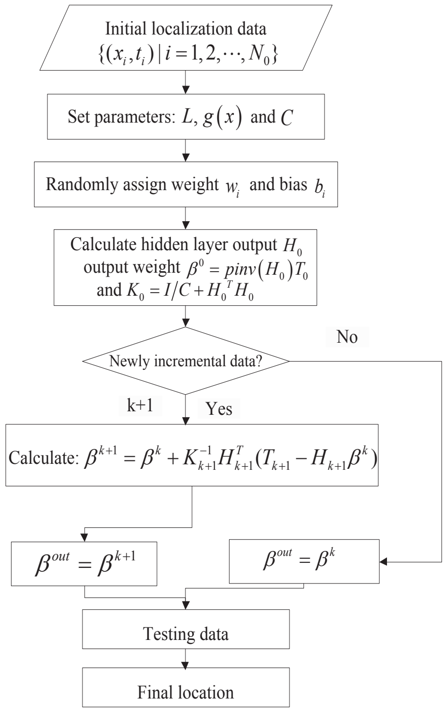


Fig. 2. Framework of COSELM

Steps for COSELM based wireless indoor localization are summarized as follows:

1) Prepare initial training data, and normalize the input features into [0,1];
2) Determine the parameters of COSELM including activation function $g(x)$, number of hidden neurons L and the regularization penalty C;
3) Randomly assign the input weight w_i and bias b_i;
4) Calculate the hidden layer output H_0 by (4);
5) Calculate the initial model parameter $\beta^0 = K_0^{-1} H_0^T T_0$, where $K_0 = \frac{I}{C} + H_0^T H_0$;
6) When the $(k+1)^{th}$ online data X_{k+1} arrives, calculate hidden layer output H_{k+1} by (4);
7) Update the model by (13) :

$$\beta^{k+1} = \beta^k + K_{k+1}^{-1} H_{k+1}^T \left(T_{k+1} - H_{k+1}\beta^k\right)$$

$$K_{k+1} = K_k + H_{k+1}^T H_{k+1},\ K_0 = \frac{I}{C} + H_0^T H_0$$

8) Calculate the estimated locations when testing data are put into the system.

Step 6) to 7) is the online updating procedure to get new model, so that the localization system can run long time with desirable performance in highly dynamic indoor environment. The framework of proposed COSELM is illustrated in Fig.2.

IV. EVALUATION

The performance of proposed COSELM is validated in real indoor environment, which is a 15m*10m working area on the 8th floor of a research building. The layout of this environment is shown in Fig. 3. Circle points are locations where data are collected. The reason to choose this test bed is that it is a typical indoor environment, and peoples' movement and other changing elements lead to the high dynamics. Wi-Fi data are casually collected at arbitrary time of days with smart phones for two months, and the collection distance is 0.5~2m. There are 32 stable APs in the environment, hence, each input vector (fingerprint) is a 32 dimension vector, and each dimension's value is the RSS of AP.

The localization performance of difference algorithms is evaluated from localization accuracy and time consumption. The experimental configuration is as follows: DELL PC with 32 bit Operation System, 2.00G RAM, Intel Core2 CPU, and Matlab R2009b.

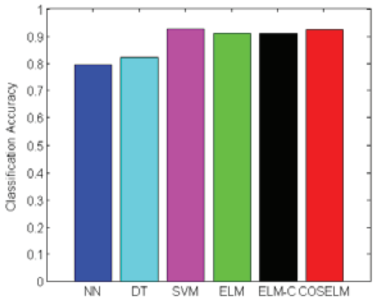
A. Comparison with other localization algorithms

To show the effectiveness of proposed online learning method, we first compare the performance of COSELM with some existing batch learning localization methods like: NN [7], DT [8], SVM [9], ELM [16] and ELM-C [17].

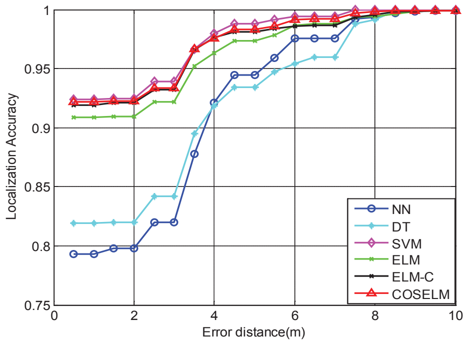
Here the total dataset contains 2400 training data and 1050 testing data. For batch learning methods, all the training data are used to get the model at one time; for COSELM, 1000 training data are used as initial data, and the rest training data are incremented for 8 times. The classification accuracy is illustrated in Fig.4(a). All the algorithms adopt sigmoid ("sig") function as activation function. Other parameters for different algorithms are: 1) SVM: $c = 2^3, g = 1$; 2) ELM: 500 hidden neurons; 3) ELM-C: 1500 hidden neurons, $C = 2^{12}$; 4) COSELM: 1500 hidden neurons, $C = 2^{12}$.

In Fig.4(a), the proposed online learning algorithm COSELM achieves 92.19% classification accuracy, which is similar to SVM (92.38%), ELM (90.86%), ELM-C (91.90%), and better than NN (79.33%), DT (81.90%). Even though COSELM utilizes online data to gradually update the model, it performs no worse than batch learning methods. And with the increase of error distance, localization accuracy (localization accuracy is the percentage that the number of right localized spots accounts for the total number of testing data, and right localized spot indicates that the distance between estimated location and expected location is smaller than error distance) for each algorithm improves, and COSELM still has the similar performance to SVM, ELM, ELM-C, and better than NN and DT, which is shown in Fig.4(b).

Fig. 3. Layout of indoor environment



(a) Classification accuracy

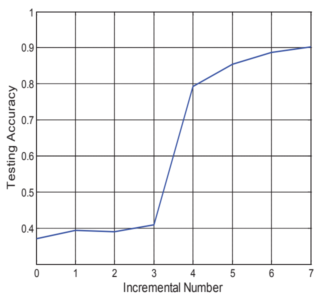


(b) Localization accuracy

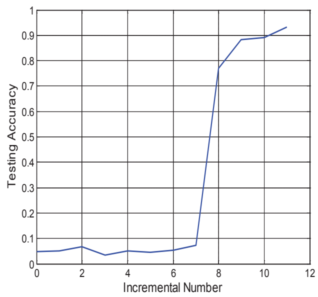
Fig. 4. Localization accuracy for different algorithms

TABLE I. TIME CONSUMPTION

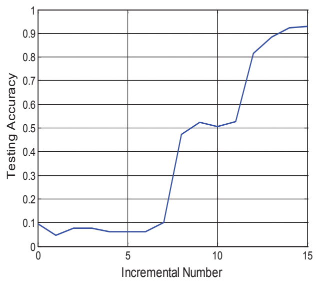
	NN	DT	SVM	ELM	ELM-C	COSELM
Training Time(s)	0	0.875	2.23	2.25	3.12	1.57
Testing Time(s)	14.26	0.875	0.93	0.09	0.25	0.25



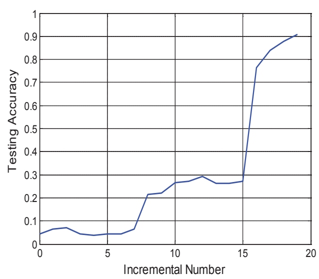
(a) Incremental number=7



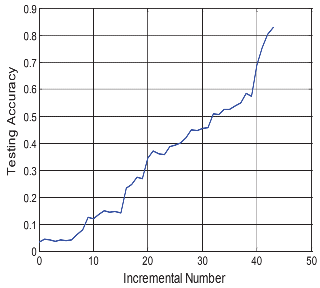
(b) Incremental number=11



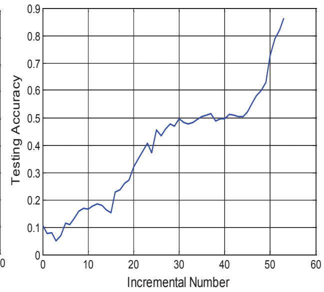
(c) Incremental number=15



(d) Incremental number =19



(e) Incremental number=43



(f) Incremental number=53

Fig. 5. Localization accuracy with different incremental number

Localization accuracy is one important measurement index for localization method, and time cost is another important one. The time consumption for different algorithms is shown in Table I. From Table I, we can see that ELM has the least testing time consumption. ELM-C and COSELM cost more time than ELM, which is reasonable, for they have bigger number of hidden neurons than ELM. ELM related algorithms' (ELM, ELM-C, and COSELM) cost less time than SVM and DT, because their input weight and bias are randomly assigned. NN costs most of the time, for it has to calculate similarity (dissimilarity) sample by sample which leads to its $O(M_1 \times M_2)$ time complexity, where M_1 and M_2 are the number of training data and testing data. Considering the tradeoff between localization accuracy and time consumption, COSELM has relatively better performance.

B. Performance with time changing

As claimed that Wi-Fi signal fluctuates severely over time due to the highly dynamic environment. To validate the localization performance along the timeline, COSELM is compared with incremental learning method OSELM and batch learning methods ELM and ELM-C. We only compare COSELM with one online learning method OSELM, for the greater effectiveness of OSELM than other sequential learning methods has been shown in [18]. The experiment is set as follows: At first, 200 data are collected to set up an initial model; then incremental data are gathered to update the existing model; finally, testing data are obtained to validate the localization performance.

The result of COSELM is shown in Fig. 5. Each of the six figures is the performance of COSELM when the incremental number is 7, 11, 15, 19, 43 and 53 respectively. The incremental number indicates how many times the model has been updated, when incremental number equals 0, it means there is no update of model, only initial model. Usually the incremental data arrive along timeline. For all these six figures, X axis represents the incremental number; Y axis is the localization accuracy when the error distance is 1m. All these figures have the same trend, which is localization performance improves when the incremental number increases. This phenomenon is reasonable for two reasons. One reason is that the total training dataset is enlarged by newly incremental data, so the dataset can cover more possibilities for fingerprints at the same location. The other reason is the newly updated model can reflect current indoor environment better.

TABLE II. ACCURACY COMPARISON

Algorithm / Accuracy	ELM	ELM-C	OSELM	COSELM
7th Increment	89.80%	91.58%	87.76%	90.31%
11st Increment	91.29%	92.13%	88.20%	**93.26%**
15th Increment	90.98%	92.23%	88.97%	**92.98%**
19th Increment	90.23%	90.73%	83.96%	**90.73%**
43rd Increment	82.75%	81.75%	67.25%	**83.00%**
53rd Increment	84.89%	84.89%	74.31%	**86.40%**

From the figures, it is obvious when the incremental number is small (which means the model was learnt long time ago and not fitted for current environment), the localization accuracy is unacceptably low like 10%-50%, which testifies that the fluctuation of RSS with time passing by makes the old model function badly for newly collected data. From these six figures, the accuracy of the last incremental number (the newest model) is the performance of current localization model, which is 90.31%, 93.26%, 92.98%, 90.73%, 83.00%, and 86.40% respectively; most of the accuracies stay in the range 85%-90%. Therefore, during the two months period, the system can perform robustly with desired accuracy, which makes a lifelong and high accuracy system viable.

After validating the localization accuracy of COSELM, we compare its performance with other three algorithms, which is shown in Table II. The parameters are as follows: the number of hidden neurons for ELM-C, OSELM, COSELM and ELM in the last two rows is 1000, the number of hidden neurons for ELM in the front four rows is 500; the constraint parameter for ELM-C and COSELM is 2000. From Table II, COSELM has similar performance to ELM-C, and better accuracy than OSELM. The effectiveness of the constraint parameter of COSELM is obvious when the incremental number is 53 that COSELM improves the accuracy by 11% than OSELM.

Table III shows the time consumption for all these algorithms. For each algorithm, it has two columns of time consumption, the first column is training time cost, and the second column is testing time consumption. Compared the time cost of OSELM and COSELM, apparently they are similar, which means the proposed method does not increase the calculation burden. In contrast to batch learning ELM, the advantage of online incremental learning is obvious; when the incremental number is 53, COSELM only costs 0.56s to update the model, while ELM needs 29.25s to retrain the model with all the training data. ELM costs more time than ELM-C when they have the same number of hidden neurons and input data, because the matrix inversion procedure of ELM is time-consuming.

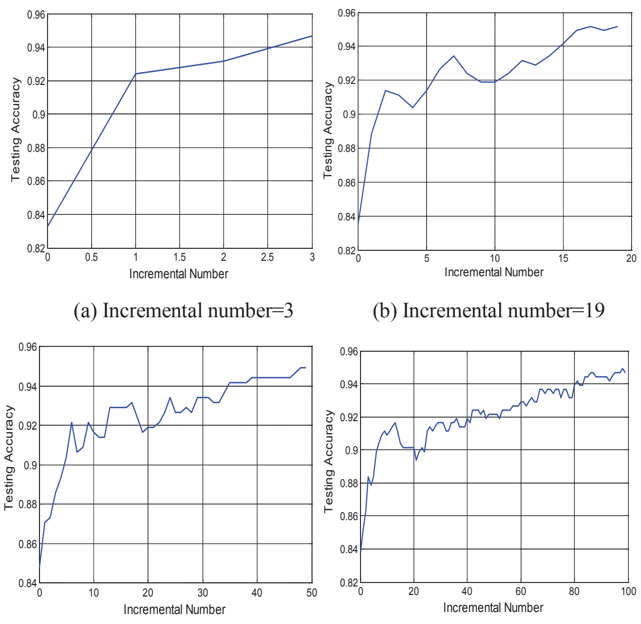
For incremental learning, the numbers of initial data and incremental data are the variables that may affect the performance of algorithm. Therefore, these two variables' effects are studied. The impact brought by the number of incremental data is illustrated in Fig.6. The experiment is implemented on the dataset with 800 training data and 395 testing data. Here the number of initial data is 200; the rest training data are incremental data. From Fig.6(a) to Fig.6 (d), the incremental number changes as: 3->19->49->99.

From the four figures, the initial localization accuracy (when increment number is 0) of each situation is similar, for they have the same initial data. And it is also obvious that no matter what the incremental number is, the final localization accuracy is still similar. These figures illustrate that, when the initial data are fixed, and the total incremental data are the same, the incremental way will not affect the final result.

The influence of initial data is also validated on the same dataset as above. The number of initial data changes as: 100->200->300->500, the rest training data are incremental data for four times, and the result is shown in Fig.7. With the number of initial data increase, the initial localization accuracy becomes better, which is 69.62%->85.32%->90.88%->92.91%. It is understandable for larger dataset leads to better result. However, the final localization accuracy for these four circumstances is almost the same considering that the total number of training data is the same.

TABLE III. TIME CONSUMPTION

Algorithm / Time Cost(s)	ELM		ELM-C		OSELM		COSELM	
7th Increment	2.19	0.03	1.15	0.06	0.57	0.06	0.57	0.06
11st Increment	2.52	0.03	1.61	0.06	0.57	0.06	0.57	0.06
15th Increment	2.80	0.03	2.35	0.07	0.56	0.06	0.57	0.06
19th Increment	3.52	0.03	2.58	0.06	0.57	0.06	0.57	0.06
43rd Increment	27.17	0.09	5.55	0.06	0.57	0.06	0.56	0.06
53rd Increment	29.25	0.09	6.50	0.06	0.56	0.06	0.56	0.06



(a) Incremental number=3
(b) Incremental number=19
(c) Incremental number=49
(d) Incremental number=99

Fig.6. Localization accuracy with different incremental number

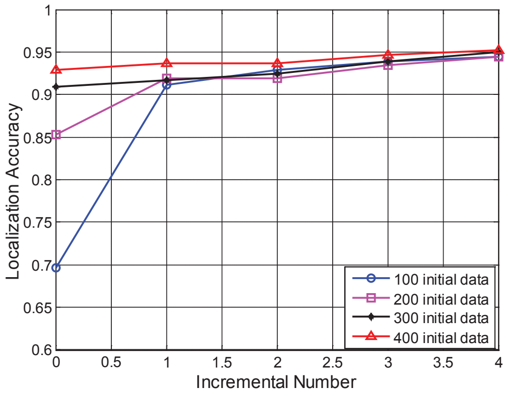


Fig. 7. Localization accuracy with different number of initial data

Combining the conclusions of initial data number's effect and incremental number's effect, we find that as long as the total training dataset is fixed, the number of initial data and the incremental number will not greatly affect the final performance. However, the larger the initial dataset is, the better the initial performance is. If the number of training data is not fixed, the more the initial data and incremental data are, the better performance will be.

V. Conclusion

Indoor localization system faces the lifelong running problem, because highly dynamic indoor environment makes Wi-Fi signal fluctuate severely. In order to solve this problem, COSELM wireless localization method is proposed, utilizing online incremental data to update the out-of-date model. The performance of COSELM is validated in real indoor wireless environment; compared with batch learning, it has similar or better localization performance, but saves much more time. Because COSELM just adopts incremental data to update model instead of retraining new model with all the training data; in contrast with original OSELM, COSELM has better performance because the L2 regularization guarantees the generalization ability and enlarges the value space of C to achieve best performance. The two months' experimental result shows that COSELM can maintain the localization accuracy in the range of [80%, 95%] with less time consumption, which makes a longtime running localization system viable.

Acknowledgment

This work is supported by National Natural Science Foundation of China (NSFC) General Program under grant No.61173066, Major Research Plan Project No.90820303, and Guangdong Province Strategic Cooperation Project of the Chinese academy of sciences No. 2011A090100001.

References

[1] N. Chang, R. Rashidzadeh, and M. Ahmadi, "Robust indoor positioning using differential wi-fi access points," IEEE Transactions on Consumer Electronics, vol.56, no.3, pp. 1860-1867, 2010.

[2] Q. Yang, S.J. Pan, and V.W. Zheng, "Estimating location using Wi-Fi," IEEE Intelligent Systems, vol. 23, no.1, pp. 8–13, 2008.

[3] F.S. Jiménez, Prieto C, "A survey of mathematical methods for indoor localization," 2009 IEEE International Symposium on Intelligent Signal Processing, 2009, pp. 9-14.

[4] Y.C. Chen, J.R. Chiang, H.H Chu, P. Huang, and A.W. Tsui, "Sensor-assisted Wi-Fi indoor location system for adapting to environmental dynamics," Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '05), 2005, pp. 118-125.

[5] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), 2004, pp. 14-23.

[6] N. Fet, M. Handte, and Pedro Jose Marron, "A model for WLAN signal attenuation of the human body," Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, 2013, pp. 499-508.

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

[8] J. Yim, "Introducing a decision tree-based indoor positioning technique," Expert Systems with Applications, vol. 34, pp.1296–1302, 2008.

[9] L.B.D. Mundo, R.L.D. Ansay, C.A.M. Festin, and R.M.Ocampo, "A comparison of Wireless Fidelity (Wi-Fi) fingerprinting techniques," 2011 International Conference on ICT Convergence (ICTC), 2011, pp. 20-25.

[10] B.B. Parodi, H. Lenz, A. Szabo, H. Wang, J. Horn, J. Bamberger, and D. Obradovic, "Initialization and online-learning of RSS maps for indoor/campus localization," IEEE Symposium Position Location and Navigation, 2006.

[11] J. Liu, Y. Gu, Y. Chen, and Y. Cao, "Incremental localization in WLAN environment with timeliness management," Chinese Journal of Computer Sciences, vol.36, no.7, pp. 1448-1455, 2013. (In Chinese)

[12] Z. Sun, Y. Chen, J. Qi, and J. Liu, "Adaptive localization through transfer learning in indoor Wi-Fi environment," 2008 Seventh International Conference on Machine Learning and Applications (ICMLA), 2008, pp. 331-336.

[13] V.W. Zheng, E.W. Xiang, and Q. Yang, and D. Shen, "Transferring localization models over time," the Twenty-Third AAAI Conference on Artificial Intelligence, 2008, pp.1421-1426.

[14] C. Koweerawong, K. Wipusitwarakun, K. Kaemarungsi, "Indoor localization improvement via adaptive RSS fingerprinting database," 2013 International Conference on Information Networking (ICOIN), 2008, pp. 412-416.

[15] C.C. Lo, L.Y. Hsu, Y.C. Tseng, "Adaptive radio maps for pattern-matching localization via inter-beacon co-calibration," Pervasive and Mobile Computing (PMC), vol. 8, pp. 282–291, 2012.

[16] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," 2004 IEEE International Joint Conference on Neural Networks, pp. 985-990, July 2004.

[17] G.B. Huang, H. Zhou, X. Ding, R. Zhang, "Extreme Learning Machine for regression and multiclass classification," IEEE Transactions on Systems, Man, and Cybernetics, vol. 42, no. 2, pp. 513-529, 2012.

[18] N.Y. Liang, G.B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate on-line sequential learning algorithm for feedforward networks", IEEE Transactions on Neural Networks, vol. 17, no. 6, pp. 1411-1423, 2006.