A Modified KNN Indoor WiFi Localization Method With K-median Cluster

IOP Conf. Series: Materials Science and Engineering **608** (2019) 012008 doi:10.1088/1757-899X/608/1/012008

A Modified KNN Indoor WiFi Localization Method With K-median Cluster

Wei Lan[1] **and Hongxin Li**[1]

[1]School of Information Science and Engineering, Lanzhou University, Lanzhou, 730000,China

E-mail :396820924@qq.com

Abstract: Because of the serious attenuation and multi-path effect of GPS signal, outdoor-positioning technology can not be applied in complex indoor environment. Through the study of K-Nearest Neighbor applied in WiFi positioning, according to the problem that the time complexity of KNN algorithm increases linearly with the quantity of samples, this paper combined clustering algorithm with KNN optimized the similarity measure in fingerprint feature space and proposed a efficient indoor target location algorithm . Experimental results showed that the algorithm improved the positioning accuracy, had strong robustness to noise and more importantly, the positioning time was effectively shortened and it can meet the requirements of practical applications.

1. Introduction

Positioning technology in indoor environment is widely used in logistics management, smart home, modern industry, etc. GPS target location and tracking technology in outdoor environments is developing rapidly and with high positioning accuracy. However, GPS is not well suited for use in closed environments due to limited signal propagation. As a result, a large number of new positioning methods have emerged in this field. The infrared positioning and tracking technology in the literature [1] is susceptible to external interference, and the infrared radiation distance is short, which is not suitable for indoor environments. Although the ultrasonic positioning tracking proposed by the literature [2] has high tracking accuracy and wide positioning range, it is not put into use a lot due to high cost, and it has certain harm to human health for a long time. Document [3] RFID positioning technology proposed by expensive hardware also failed to be accepted by the public. WiFi location tracking technology is a technology that uses WiFi hotpots for target location and tracking. It can locate existing infinite LANs in indoor environments. The existing smart devices are generally equipped with WiFi functions, which do not require additional hardware costs, and the accuracy can meet the requirements, which has been widely concerned by researchers at home and abroad.

WiFi positioning can be divided into three types: (1) Location-based radio wave arrival time; (2) Location-based radio wave arrival angle; (3) Intensity localization based on wireless wave arrival signal[4]. Considering the difficulty of angle and time measurement during radio wave propagation, this paper only discusses the location tracking algorithm based on received signal strength(RSS). Depending on the algorithm used, positioning techniques can be divided into two categories: Positioning based on a signal propagation model and the position location fingerprinting algorithm. The former is modeled through the channel of the spread of WiFi, but the complexity and diversity of the indoor environment is often difficult to achieve. The latter is by constructing the signal strength

and location information database match, the best match and the predicted position by the evaluation function, the feasibility of the former is far superior.

The existing KNN method has two problems: large computational complexity and sensitivity to RSS signal noise. Aiming at these two problems, this study builds a fingerprint database, selects the k-median clustering algorithm to cluster the samples, reduces the algorithm overhead, optimizes the similarity measure, and proposes a WiFi localization algorithm based on k-median-WKNN. Experiments were conducted in the laboratory to further compare and analyze the positioning results.

2. WiFi fingerprinting algorithm based on location fingerprint

2.1. RSS

RSS is defined as: received signal strength [4-7]. RSS is generally negative, and the stronger the signal, the larger the value. Ideally, the measured RSS values determined under the same general location of the device is substantially unchanged. Because of the complexity of the radio propagation, the same position measured RSS values there is a large deviation of the actual application, and the change with time of the RSS value is also evident, but there may be missing signals in the plurality of sets of data. The arrow in Figure 1 shows a large error point.

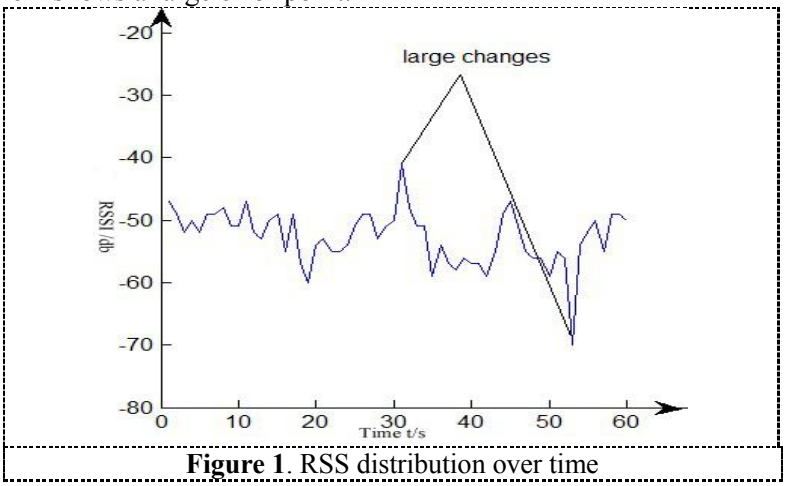

Figure 1. RSS distribution over time

2.2. WiFi localization algorithm based on K-nearest neighbor method

K-nearest neighbor method is widely used in the field of data analysis. The algorithm to find the minimum distance from the target sample with the *k* reference points in the feature space. The properties of the target sample are determined by discriminating the properties of the K reference points. Assume that the experimental site selects θ WiFi hotpots. A total of α samples are collected at each location to form a location fingerprint database, and the data training set is:

$$K = \{(r_1, p_1), (r_2, p_2), \cdots, (r_\alpha, p_\alpha)\} \tag{1}$$

Vector $r_i = (r_{i1}, r_{i2}, \cdots, r_{i\theta}) \in R^\theta$ in the formula means that RSS signal measured by θ access points. r_{i1} represents the data collected from the first AP. r_{i2} represents the data collected from the second AP. From this analogy, it can be concluded that $r_{i\theta}$ represents the data collected from the θ Vector $p_i = (x_i, y_i) \in R^2$ represents the corresponding position vector, where $i = 1,2,\cdots,\alpha$.

During the online positioning phase, a set of RSS signals can be measured at the measurement location. That is, the r vector is known, and the position P vector is estimated. First, through Equation 2, *k* reference points closest to the distance of the r vector in the location fingerprint library feature space are calculated. The distance between the r vector and the training sample r_i is defined as:

$$d = [\sum_{\beta=1}^{\theta} |r_{t\beta} - r_{i\beta}|^q]^{\frac{1}{q}} \quad (2)$$

Where q is a positive integer and Manhattan distance when $q=1$. Euclidean distance when $q=2$. Text uses Euclidean distance to select neighbors. $(x_{k1}, y_{k1}),(x_{k2}, y_{k2}),\cdots,(x_{kk}, y_{kk})$ are its coordinates. Taking the average of these coordinates to get the positioning result is (x, y).As can be seen from the above, the KNN algorithm needs to calculate the Euclidean distance for each training sample, which does not need to be considered for the small sample set, but when the sample size of the fingerprint database reaches a certain level, the complexity of the algorithm increases, and the practical application is therefore limited.

3. Improvement of WiFi Location Algorithm Based on K-median-WKNN

3.1. Algorithmic process

The improved algorithm in Figure 2 corresponds to two processes: the offline learning phase and the online positioning phase. In the offline phase, by collecting the RSS signal and position coordinates, after the data is cleaned, the location fingerprint database is constructed, and then the samples are K-median clustered, and each sample is grouped into one cluster. In the online reasoning stage, the class to which the verification sample belongs is first defined, and the sample points are weighted and averaged to obtain the observation position.

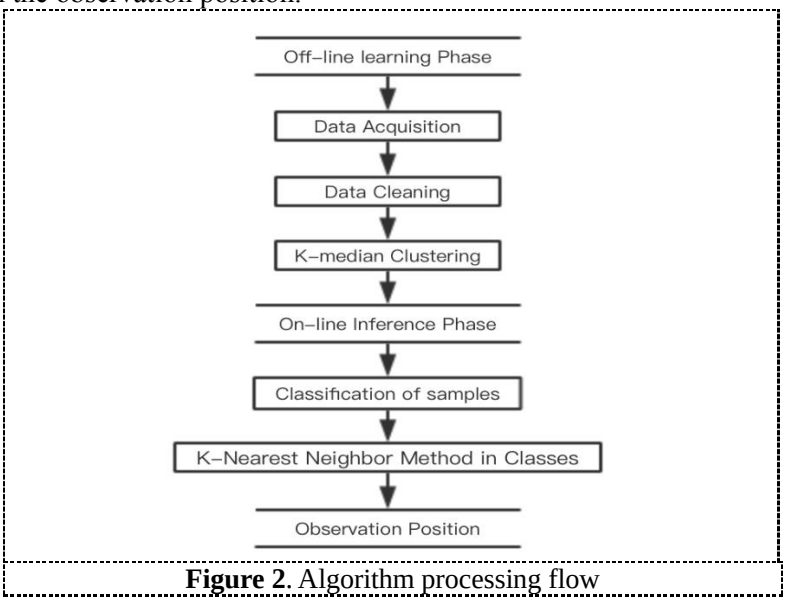

Figure 2. Algorithm processing flow

3.2. Selection of clustering algorithm

The K-means clustering algorithm proposed in [5] can quickly cluster the location fingerprint database of the localization area. However, in the WiFi positioning process, the method of averaging the RP point RSS information to update the class center vector often has a large error. For example, the RSS value collected by the same AP at a certain location is(-41,-42,-42,-43,-89). Due to the presence of noise points (outliers), and the mean value of -51 with the actual RSS there is a large variation, due to the complexity of the indoor environment, such noise often unavoidable point.

The K-medoids clustering proposed in [6] updates the class center vector by selecting the point with the smallest extinction error of each sample point distance in the cluster, which avoids the distortion of the class center vector caused by the noise point, and the corresponding cost is It is an increase in time complexity. Considering the particularity and comprehensive requirements of indoor location fingerprint location, this study selects K-median clustering algorithm. The difference between

this algorithm and K-means is that when updating the class center vector, the mean value is used instead of the mean value to avoid the noise point. Distortion, while inheriting the fast and simple features of K-means.

The approach to clustering is to measure the similarity between the assignments assigned to each cluster to minimize a specified cost function. The cost function in this experiment is:

$$J = \sum_{n=1}^{\alpha} \sum_{k=1}^{\delta} \left\| r_n - E_k \right\|^2 \tag{3}$$

Where r_n is the RSS vector for each location point, E_k was r_n center vectors of their class. k is the number of clusters. n represents the number of RSS vector samples. The essence of K-median lies in the problem of minimizing the total cluster variance J by continuous classification. The selection of the cluster number K is particularly important.

3.3. Positioning algorithm improvement

3.3.1 Offline learning phase. In the offline stage, first of all training data K-median clustering. Assume that the number of samples in the fingerprint database is α, each sample point receives the RSS value from θ APs, divides α reference points into δ clusters, and defines reference point c to receive the RSS value from the θ AP as $r_{c\beta}(c = 1, \cdots, \alpha; \beta = 1, \cdots, \theta)$. The degree of dissimilarity between sample point a and sample point b is defined as:

$$d = [\sum_{\beta=1}^{\theta} \left| r_{a\beta} - r_{b\beta} \right|^2]^{\frac{1}{2}} \tag{4}$$

K-median clustering of samples in the fingerprint database, the clustering process consists of three steps:

Step 1: Select the initial class center reference point.

(1) Randomly select a sample point A as the first class center reference point, calculate the distance d_A from the remaining points to the sample point A. The introduction of the weighting factor $1 - 1/d_A$ makes the probability that the farther point to A is selected is greater.

(2) On the basis of the above, select the second class center reference point B, and calculate the distance $\{d_A, d_B\}$ between the remaining points to the sample points A and B. the greater the probability that the point farther away from the distances A and B is selected.

(3) Repeat the above two steps until the δ class center reference points are selected.

Step 2: Update the class center reference point.

The new class center reference point is re-determined in each class by re-determining the new class center vector by taking the median value for each dimension of the sample RSS vector $r_i = (r_{i1}, r_{i2}, \cdots, r_{i\theta})$ in the class in each class.

Step 3: The reference point is reassigned to the class center reference point.

(1) Calculate the distance from each sample point to all class center vectors and assign them to the nearest class.

(2) Repeat the second step until the class center vector changes within the threshold and the algorithm stops.

3.3.2 Online positioning stage. The purpose of this algorithm is to estimate its approximate position vector (x, y) in the fingerprint database by measuring a target sample $J = (r_1, r_2, \cdots, r_\theta)$ representing the signal strength from θ APs.

Step 1: Delineate the class to which the point to be located belongs.

Calculate the similarity between the J vector and various center vectors $\{E_1, E_2, \cdots, E_\delta\}$, and delineate the cluster to which the J vector belongs. The cosine similarity function proposed in [7] only pays

attention to the directional difference of vectors, ignoring the difference in vector amplitude. Therefore, this study introduces Tonimoto coefficients to describe the differences between vectors. The similarity function of the *J* vector and the central vector Ei of the *i* class is defined as:

$$TonSim_i = \frac{E_i \cdot J}{\|E_i\|^2 + \|J\|^2 - E_i \cdot J} \tag{5}$$

Step 2: Determine the spatial coordinates in the class to which the point to be located belongs. In this class, the similarity between the J vector and the S training sample RSS vectors in the class is calculated. The similarity function is shown in Equation 5. After the ascending order, select the similarity of the K training samples $\{r_{m1}, r_{m2}, \cdots, r_{mk}\}$ with similar similarities:

$$\{TonSim_{m1}, TonSim_{m2}, \cdots, TonSim_{mk}\} \tag{6}$$

The corresponding coordinates are:

$$\{(x_{m1}, y_{m1}), (x_{m2}, y_{m2}), \cdots, (x_{mk}, y_{mk})\} \tag{7}$$

Considering that the similarity large vector has a great influence on the sample vector prediction result, the introduction similarity normalization weight $\{w_{m1}, w_{m2}, \cdots, w_{mk}\}$ is defined as

$$w_{mi} = \frac{TonSim_{mi}}{\sum_{i=1}^{k} TonSim_{mi}} \tag{8}$$

The coordinate position is then weighted according to Equation 9 to obtain its physical coordinates.

$$\begin{cases} x = w_{m1}x_{k1} + w_{m2}x_{m2} + \cdots + w_{mk}x_{mk} \\ y = w_{m1}y_{k1} + w_{m2}y_{m2} + \cdots + w_{mk}y_{mk} \end{cases} \tag{9}$$

4. Experimental result

4.1. Data collection and processing

The experimental site is a closed indoor environment of 12m × 10m. As shown in table 2, five AP points are arranged at the indoor edge position, and 120 position points are selected, and each position point collects 70 RSS data separately at different angles in the upper and lower afternoons. Figure 3 shows that the indoor environment has a large flow of people and there are many obstacles. The signal propagation has multipath effect, so signal noise is unavoidable. The RSS signal error in actual WiFi positioning is the key for researchers to consider. The data cleaning in this study mainly deals with two aspects:1. Missing data processing, assuming that r_{ti} is missing data, the data at time t_i is equal to the data at the previous sampling time. 2. Abnormal data processing, when the RSS value of two adjacent moments is greater than the constant 20db, the data is considered abnormal. Let $r_{ti} = r_{t(i-1)}$ .After the data is cleaned, 6021 more stable data are finally obtained. According to the idea of cross-validation, these data are divided into 8,78 estimated subsets and 1,204 verified subsets according to 8:2.

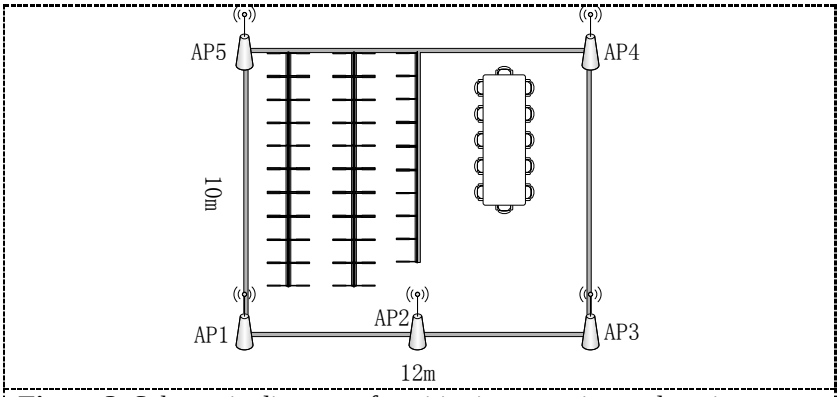

Figure 3. Schematic diagram of positioning experimental environment

4.2. Determination of the number of clusters and the number of neighbors

The value of the number k_n of the K-median cluster has a great relationship with the training samples. In this experiment, $k_n = 4$ is used after repeated measurements. The training samples are divided into four clusters and there are four class center RSS vectors. Then the target sample is delineated into the most similar cluster, and the WKNN algorithm is used in the cluster. The number k_n of the nearest neighbors in the algorithm needs to be artificially given. In this experiment, the error results of the two algorithms are obtained by measuring different k_n selections. As shown in Figure 4:

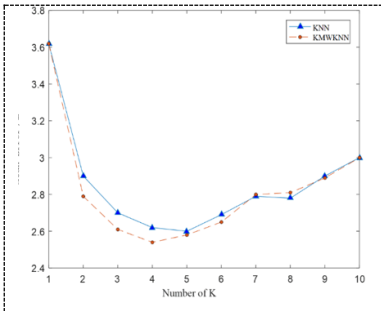

Figure 4.Relation between error and Kn

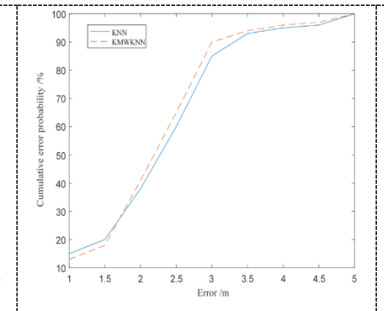

Figure 5. Comparison of error accumulation probability

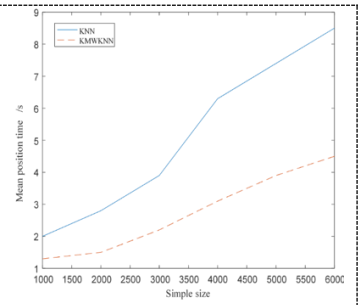

Figure 6. Comparison of error accumulation probability

It is not difficult to find from the figure that the value of k_n is too small, the sample is susceptible to noise interference, and the error of observation is larger. When the value of k_n becomes larger, the effect of noise becomes smaller and smaller, and the error also educed. However, when the value of k_n is too large, samples with lower similarity also participate in the weighting of the sample, which in turn leads to an increase in the error of the sample. It is not difficult to see that $k_n = 4$ is the best in this experiment. As can be seen from the above figure, compared with the traditional KNN algorithm, the improved algorithm uses K-median clustering to reduce the sample size, but the difference between the two is less than 0.3m. In this experiment, $k_n = 4$ was taken and the experimental results were further observe.

4.3. Comparison before and after algorithm improvement

Figure 5 shows the cumulative probability of different positioning errors of the two algorithms. The former can be seen that the positioning error of the improved algorithm starts to converge within 3m, the probability of error within 2.5m is close to 80%, and the accuracy is slightly better than the KNN algorithm. The accuracy of domestic and foreign research results is in line with practical applications.

Although the accuracy improvement is not obvious, it can be clearly seen in Figure 6. As the sample size increases, the improvement of the positioning time of the improved algorithm becomes more and more significant. The larger the sample size, the more time the traditional KNN algorithm takes. Meet the requirements of indoor positioning real-time.

As shown in Table 1, this experiment will use the traditional KNN method, Deep Belief Network (DBN) algorithm. K-means-KNN algorithm and K-median-WKNN improved algorithm run on the same configuration server, and comprehensively evaluate the difference in performance of the three algorithms through five indicators. It can be seen that the DBN algorithm is significantly better than others in accuracy. The improved K-median-WKNN algorithm is slightly better than the KNN method, but it is not obvious. Therefore, it is often limited in practical applications. At the same time, the clustering algorithm improves the efficiency of the algorithm to some extent. From the perspective of error variance, K-means clustering is susceptible to interference from noise points, resulting in large fluctuations in positioning results. On the whole, the improved algorithm has certain robustness while ensuring the positioning accuracy, and shortens the positioning time, which is in line with practical applications.

Table 1. Comparison of algorithm performance

Index	KMWKNN	K-means-KNN	KNN	DBN
X axis error/m	1.41	1.45	1.56	1.17
Y axis error/m	2.02	2.01	2.05	1.53
Error mean/m	2.46	2.56	2.62	1.81
Error variance	1.31	1.54	1.61	1.15
Mean time/S	4.2	4.17	8.54	35.63

5. Conclusion

By constructing the location fingerprint database, k-median clustering is used to improve the time complexity of the algorithm, and the Tonimoto coefficient is applied to characterize the similarity between RSS vectors. A new WiFi localization algorithm based on k-median-WKNN is proposed. Theoretical derivation and experimental results show that the algorithm shortens the positioning time under the condition of ensuring accuracy, effectively suppresses the disturbance of the noise to the positioning result, and can meet the requirements of indoor positioning. Future research will focus on further optimization of the position fingerprint database and improve the positioning accuracy in this regard.

References

[1] LU Qi,SHU Guohua. An indoor orientation System based on single chips[J]. Microprocessors, 2006,2,66-68.
[2] YANG Yang,XIAO Jinhong,LIU Zhi.Indoor Three-Dimensional Positioning System Based On Ultrasonic[J].Jonrnal of Jilin University,2012,3(20):267-271.
[3] LI JunHuai, ZHANG GuoMou,YU Lei,ZHANG Jing, Analysis and Simulation of RFID positioning method for indoor environment[J]. Computer Engineering,2012(14): 276-279.
[4] JIA YuQing, WiFi localization algorithm based on deep learning[D]. Nanjing,Nanjing University,2014.
[5] Bai S,Wu T. Analysis of K-Means algorithm on fingerprint based indoor localization system [C] Microwave、Antenna,Propagation and EMC Technologies for Wireless Communications (MAPE),2013 IEEE 5th International Symposium on. IEEE 2013:44-48.
[6] TAO Zheng. Research on Improved WLAN Iocation Fingerprint location Algorithm Based on Chi-square Distance[D].DALIAN: Dalian University of Technology,2016.
[7] LIU Xingchuan,LIN Xiaokang,Fast Wi-Fi Positioning Algorithm Based on Clustering[J].Computer Engineering,2011,37(4):285-287.