2012 International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

Clustering algorithms research for device-clustering localization

Huang CHENG, Feng WAWANG, Rui TAO
Software School, Beijing University of Posts and Telecommunications
Beijing, China
chenghuang1990@gmail.com

Haiyong LUO
Ph.D,Senior Engineer
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
yhluo@ict.ac.cn

Fang ZHAO
Associate Professor
Software School, Beijing University of Posts and Telecommunications
Beijing, China
zhaofanglhy@yahoo.com.cn

***Abstract*—Crowdsourcing-based localization has attracted wide research concern to the metropolitan-scale positioning. However, crowdsourcing-based fingerprints collection with assorted mobile smart devices brings fingerprint confusion, which significantly degrades the localization accuracy. To solve the device diversity problem, many solutions have been raised like the Device-Clustering algorithm. Based on macro Device-Cluster (DC) rather than natural device, DC algorithm maintains less device types and slight calibration overhead. Despite high positioning accuracy, the selection of suitable clustering algorithms in DC system becomes another puzzle. In this paper, we reshape the novel Device-Clustering algorithm to enhance the indoor positioning by comparing the application of different clustering algorithms. The experimental result indicates the reliability of DC strategy in broad clustering scheme as well as the suitable locating process corresponding to distinct environment.**

Keywords- device heterogeneity; Device-Cluster algorithm; clustering algorithm;

I. INTRODUCTION

The explosive developments of mobile smart devices, social networks and location-aware applications have resulted in a global commercial awareness on the research of localization. By tracking and tapping a user’s location, the corresponding effect like navigation, hotspot search and interest sharing could find their way into people’s life [1,2].

Though the GPS tracking technology is currently the most used location-aware applications in providing directions and location places of interest in the open outdoor scenes, it is not effective in the indoor environment because of its dependency on the visible number of satellites. To realize indoor GPS-free localization, some new techniques like Wi-Fi signal have been introduced. With the Wi-Fi access point infrastructure, which has already been installed in a building, the Wi-Fi Positioning Systems can localize a consumer smart-phone with high accuracy and low power-consumption [2,3].

Among all available Wi-Fi positioning techniques, the fingerprint algorithm is most popular. After enough collection of Radio Signal Strength Indication (RSSI), this method will match the closest fingerprint in the database with corresponding observing fingerprint [4]. Nevertheless, the effect of fingerprinting-based localization greatly depends on the fingerprint density and sampling coverage. Thus to build an abounded fingerprint database for the metropolitan-scale localization, traditional expert sampling is time-consuming, intrusive [5]. As an alternative, a Wikipedia-style crowdsourcing model has been introduced to encourage volunteer users contributing fingerprints unnoticed when tagging places [6].

Crowdsourcing-based fingerprint collection could vastly solve the sampling coverage problem, but the following device heterogeneity issue has arisen. Volunteer user's assorted mobile devices bring diverse Received Signal Strength (RSS) pattern. Whenever the observing and training fingerprints belong to different devices, such variance remarkably degrades localization accuracy. Besides, this kind of heterogeneity lies not only on distinct Wi-Fi chipsets and antennas, but also on distinctive hardware driver, operating system versions, encapsulation materials and so forth [7]. Therefore, to maintain different Wi-Fi training fingerprint for every type of device is overly laborious and impractical in real-world applications.

To handle the device diversity, many schemes have been introduced to the calibration phase. Device-Clustering (DC) algorithm is just one of the most efficient schemes. Based on the assumption that distinct devices with the same RSS pattern are of the same macro class, the known devices can directly exploit the samplings from corresponding DCs while unknown observing could be linearly transferred into available RSS pattern. Then through user's feedback, new tracking fingerprint can be absorbed into the fingerprint database to augment radio map [8].

DC algorithm emphasizes more on the off-line clustering so that a decent clustering algorithm is essential. Previous DC positioning system only considers the density-based clustering algorithm, so in this paper we will discuss the effect of different clustering algorithm in device-clustering localization as well as the optimal process when processing the clustering phase in DC framework.

II. RELATED WORK

A. Device heterogeneity

Common research efforts on solving device heterogeneity usually fall into the following categories.

To calibrate the observing fingerprint RSS pattern is an often case. Haeberlen et al. proposed a pairwise linear transformation between different devices to cover the heterogeneity [9]. But the linear transformation only works well with uniformed combination of hardware and software, thus this technique fades away in widespread use [10]. Therefore calibration-free strategies are introduced by exploiting the ratio or difference between pairs of access points' RSS to measure the matching phase [11,12].

Transfer learning algorithm is another research hotspot in recent years. Treating the mapping problem as a multiple learning task, the RSS pattern is assumed to be uniformed under latent feature space [13]. Then the uniformed relationship learnt from both tracking devices and training devices in a low-dimensional space with Manifold Alignment [14]. However, the latent space is still limited to theoretical research recently and its computational complexity also obstructs practical promotion and deployment.

Device Clustering is a new solution against device heterogeneity [8]. Given ample device types offered by crowdsourcing-based training and laborious sampling overhead, DC algorithm plans to replace single device with macro DC. In previous research both hierarchical clustering and density-based clustering are applied to class distinct devices into corresponding DC, but the clustering criterion remains to be discussed. In this paper we will compare different kind of clustering algorithms and figure out the optimal processing flow to maximum the clustering effect and positioning accuracy.

B. Clustering algorithms

Clustering is to group data into the same area with similar characteristics, in which centroid-based clustering is a common method. Based on the minimum distance measure, centroid-based clustering algorithm will iteratively find a fixed number of cluster centers to embody the dataset. Lloyd's algorithm referred this method to "k-means algorithm" [15]. Unlike k-means clustering, the hard clustering method, Fuzzy C-Means (FCM) algorithm is a kind of soft clustering algorithm. In FCM, data elements can belong to more than one cluster, and associated with each element is a set of membership levels [16]. However, centroid-based clustering only finds a local optimum, and the computational overhead if commonly heavy.

Based on the pairwise relationships, Frey and Dueck published Affinity Propagation (AP) algorithm which clusters data elements via passing inter-point messages [17]. AP algorithm simultaneously considers all data points as potential exemplars. By propagating real-valued messages between pairwise data points, AP could automatically detect clusters. AP algorithm could automatically determine of number of clusters and also own a fast running time, but it's weak at determining initial exemplars and also space-wasting.

Density-based clustering is another staple clustering model, in which clusters are formed based on the density rather than radius. DBSCAN method is a kind of density-based clustering algorithm designed for discovering clusters of arbitrary shape [18]. Although DBSCAN is effective for spatial datasets, its performances obviously depends on two parameters: the maximum radius of a neighborhood and the minimum number of the data points in a cluster.

In view of above different clustering algorithms, we will research the application of three methods--FCM, AP and DBSCAN in DC localization to discuss the optimal selection of clustering algorithms and the finest clustering criterions.

III. ALGORITHM

Our algorithm framework is shown in Figure 1. This flow can be divided into three parts.

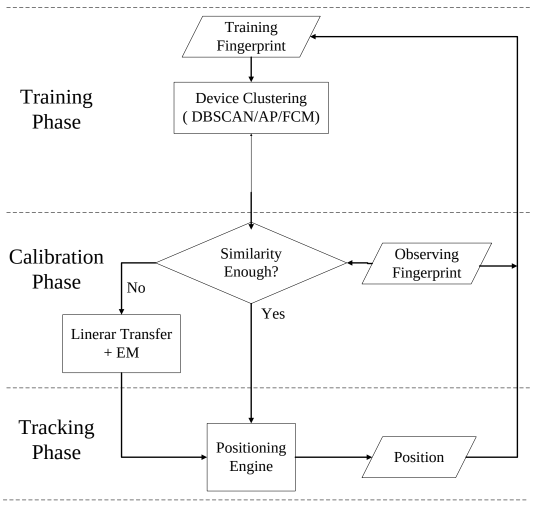

Figure 1. Algorithm Framework

In the training phase system maintains a DC database rather than single device. Whenever an observe fingerprint is input in the calibration phase, system will decide whether this device belongs to one of the DCs. If the tacking device is totally new, EM algorithm will be applied to linearly transfer the observe fingerprint. Finally in the tracking phase, the estimated position will be worked out with certain positioning algorithm.

A. Training Phase

1) Device similarity measurement

The RSS fingerprints of a specific device collected in a location not only provide information at these points, but also imply the specific characters of this type of devices. Thus by comparing these fingerprints with some measurements, similarities between pairwise devices can be obtained.

To obtain good clustering results, various clustering algorithm employs different similarity definition. In the affinity propagation clustering algorithm, the negative squared error (Euclidean distance) in fingerprints is used to measure device similarities between pair devices, as the formula (1) shows.

$$Sim(\mathbf{p},\mathbf{q}) = -\|\mathbf{p}-\mathbf{q}\| = -\sqrt{\sum_{i=1}^{d}(\mathbf{p}_i-\mathbf{q}_i)^2} \tag{1}$$

$\mathbf{p}$, $\mathbf{q}$ *represent* fingerprints of two devices, respectively, and d is their dimension, *or* the number of APs.

Contrary to the negative positive squared error, the FCM clustering algorithm employs the following positive squared error (Euclidean distance) in fingerprints to measure device similarities between pair devices.

$$Sim(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\| = \sqrt{\sum_{i=1}^{d}(\mathbf{p}_i-\mathbf{q}_i)^2} \tag{2}$$

As for the DBSCAN clustering algorithm, the Pearson Correlation Coefficient [19] is selected to measure similarities in fingerprints between pair devices.

$$r(\mathbf{p},\mathbf{q}) = \frac{\sum_{i=1}^{d}(\mathbf{p}_i-\bar{\mathbf{p}})(\mathbf{q}_i-\bar{\mathbf{q}})}{\sqrt{\left(\sum_{i=1}^{d}(\mathbf{p}_i-\bar{\mathbf{p}})^2\right)\left(\sum_{i=1}^{d}(\mathbf{q}_i-\bar{\mathbf{q}})^2\right)}} \tag{3}$$

The Pearson correlation ratio r represents the linear dependency between two fingerprint vectors. r ranges from 0 to 1, where 0 indicates the least similarity while 1 indicates the greatest similarity.

2) Fingerprint pre-processing

To obtain robust device clustering results, we employ the mean value of the training fingerprints to compute the device similarity with the above three formulas, which can decrease the effect of RSS fluctuation.

B. Calibration Phase

1) Known device calibration

By applying the crowdsourcing method to collect training samplings, different device clusters will cover different location area after device clustering. When a known device, which has been clustered into a specific class, enters into a fresh location where there is not fingerprint data corresponding to its belonging cluster in the training database, using the other existing cluster there to perform fingerprint mapping is necessary.

The known device calibration employs the best linear transformation between two clusters to map the observing fingerprint to the cluster with trained fingerprint data in a specific place, which will be used to position the device.

The best linear transformation between two clusters is defined as the fomula (4)

$$RSS_{C_2} = \mathbf{a}RSS_{C_1} + \mathbf{b} \tag{4}$$

$\mathbf{a}$, $\mathbf{b}$ *represent* the least square linear regression coefficient vector between cluster C_1 and C_2 , respectively. RSS_{C_1} and RSS_{C_2} are the RSS fingerprint data corresponding to the two clusters (C_1 and C_2).

2) Unknown device calibration

When a new observing fingerprint enters, if the system judges that the tracking device has not been trained, the below unknown device calibration process is necessary.

If $\mathbf{F}_{train}$ and $\mathbf{F}_{track}$ are fingerprint databases each from training devices and unknown tracking devices, then there should be a pair coefficient (a,b) which satisfies $\mathbf{F}_{train} = a\times\mathbf{F}_{track} + b$. In this paper, Expectation Maximization algorithm (EM) is employed to refine the linear parameter over and over by repeatedly computing expectation and maximization, until the fluctuation of linear transfer convergent [7]. In this way, the fingerprints pattern from tracking devices is transformed to training fingerprint pattern.

In the parameter-learning process, f^0 indicates a DC p's training fingerprint database, $\Theta = (a,b)$ denotes the linear coefficients and f_{qt} represents the tracking device q 's RSS readings at time t . EM starts with an initial guess of the parameters (a_0,b_0) and an initial observing fingerprint f_{q0} . Then it seeks to find the optimal linear coefficient by iteratively applying the following two steps:

E-step: Treat Θ_t as a constant, then work out the $f_{q(t+1)}$ with transformation function $T(\Theta_t, f_{qt})$ until the expected log-likelihood of the following formula is greatest:

$$SIM = \{dist(f_{qt}, f_p) \mid f_p \in f^0\} \tag{5}$$

M-step: Treat $f_{q(t+1)}$ as a constant, find Θ_{t+1} to satisfy the greatest $E(\Theta_t, f_{qt})$.

The optimization problem is as follows:

$$\underset{\Theta}{argmax}\, E(\Theta_t, f_{qt}) = \underset{\Theta}{argmax}\left\{\frac{E(\Theta_{t+1}, f_{q(t+1)})}{E(\Theta_t, f_{qt})}\right\} \tag{6}$$

When the fluctuation of linear transformation function stays within a fixed range, for example 0~1, we deem the function convergent and the final linear coefficients (a,b) could be used to map observing fingerprints.

C. Tracking Phase

Tracking phase is the final positioning phase, in which the transformed observing fingerprint and corresponding DC serve as the input to the positioning algorithm. Here we introduce the

Bayesian localization framework [20], which is very suitable for crowdsourcing-based localization for its simplicity and high accuracy.

Bayesian localization method computes the posterior probabilities over locations to find the most possible position. Given an observing fingerprint o and a random location $l_k \in L^n$ where L indicates a set of n locations, the posterior probability that o belongs to l is shown by the Bayes' rule:

$$P(l_k \mid o) = \frac{P(o \mid l_k)P(l_k)}{\sum_{i=1}^{n} P(o \mid l_i)P(l_i)} \quad (7)$$

$P(l_k \mid o)$ indicates the prior possibility of the occur of observing fingerprint. $P(l_k)$ is usually set as the uniform distribution so in practical use people often ignore it. Therefore the estimated location l_e is the one obtaining the maximum value of the posterior probability.

$$l_e = \arg\max_{l_k} P(o \mid l_k) \quad (8)$$

Supposing each $o = (v_1, v_2, ..., v_m)$ and M is the index of m access points, then the $P(o \mid l_k)$ in Equation 8 becomes:

$$P(o \mid l_k) = \prod_{i \in M} P_{v_i|L}(v_i \mid l_k) \quad (9)$$

The $P_{v_i|L}$ here is often modeled as some kind of distribution. In this paper we choose Gaussian distribution (maximum-likelihood parameter estimator) [21].

If in location l_k, there are totally n training fingerprints in one DC and each fingerprint scans m access points, we denote $\mathbf{T}_i = (s_1, s_2, .., s_n)$ as the RSS set of all the fingerprint value in access point i.

If $\mathbf{T}_i$ is treated as Gaussian distribution, then the probability function of v_i is as follows:

$$P_O(v_i \mid l_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(v_i - \mu)^2}{2\sigma^2}} \quad (10)$$

where μ and σ^2 are mean value and variance of $\mathbf{T}_i$.

IV. Experiment

A. System Setup

To compare the performance of different Device-Clustering algorithms, we deploy a simulation environment on the 7th floor of Institute of Computing Technology (ICT) in the Chinese Academy of Sciences. The sampling and positioning area is shown in Figure 2. This area covers 28 stations as is shown in figure and the red points represent the sampling locations.

In this experiment, we employ eight mobile smart devices which are further divided into six different types. The detailed device information which will affect the RSS is shown in Table 1. We carry each device to these red points for 2 minutes to collect RSS data and corresponding location flags. To rule out other impacts, the sampling time is consistent. Finally we choose 100 samples for every mobile device in each grid. We use two same types of devices (two HTC G14 smartphones and two P7500 Pads) to test whether the three algorithms can cluster them in a same cluster.

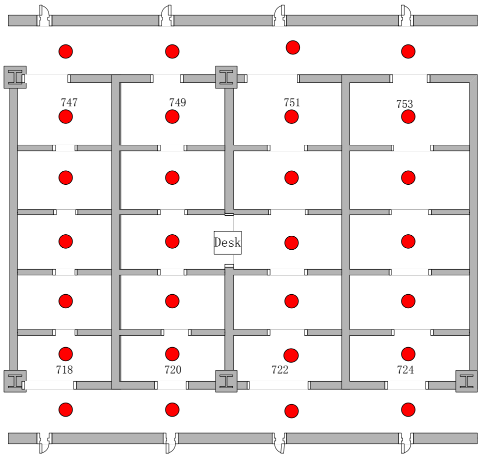

Figure 2. Sampling area on 7th floor of ICT

Fingerprinting-based positioning algorithms don't need to know the positions of different access points, so we don't care the deployment of the beacons in our simulation environment.

TABLE I. DEVICES USED FOR DATA COLLECTION

Device	Platform	Wi-Fi Module	OS
1	HTC Desire HD （G10）	Broadcom BCM4329	Android 2.2
2	HTC G14(1)	Broadcom BCM4329	Android 2.3.4
3	HTC G14(2)	Broadcom BCM4329	Android 2.3.4
4	Samsung Galaxy S（i9000）	Samsung SWB-B23	Android 4.0
5	MIONE 1S（M1S）	Broadcom BCM4329	Android 4.0
6	Huawei U8860	Unknown	Android 4.0
7	P7500(1)	Broadcom BCM4330	Android 3.1
8	P7500(2)	Broadcom BCM4330	Android 4.0

B. Clustering effect Evaluation

In this experiment, we adopt AP, FCM, and DBSCAN algorithms to cluster the 8 devices, respectively.

As mentioned above, in the AP clustering algorithm, the negative squared error (Euclidean distance) in fingerprints is used to measure device similarities between pair devices. Table II shows the similarities between devices in Euclidean distance, while Table III shows the AP clustering results. As we expect that AP algorithm clusters two P7500 pads into one group, two G10 smartphones into another group, and the left devices into third group.

TABLE II. SIMILARITIES BETWEEN DEVICES IN EUCLIDEAN DISTANCE

Device	1	2	3	4	5	6	7	8
1	0	801	710	653	310	752	1251	2075
2	801	0	245	294	333	405	450	734
3	710	245	0	253	428	362	367	767
4	653	294	253	0	229	117	240	294
5	310	333	428	229	0	398	515	1123
6	752	405	362	117	398	0	323	659
7	1251	450	367	240	515	323	0	200
8	2075	734	767	294	1123	659	200	0

TABLE III. CLUSTERING RESULT WITH AP

Device	1	2	3	4	5	6	7	8
DC	1	2	2	2	2	2	3	3

FCM algorithm employs the membership value which adds weight to the distance from one vector to each cluster, as shown in Table IV. From the membership matrix we derive the final clustering results shown in Table V which is a little different form the results in AP algorithm. FCM algorithm also clusters two P7500 pads into one group, G10 and M1S into one group (which is different from the clustering result with AP), and the others into one group.

TABLE IV. FINAL MEMBERSHIP MATRIX IN FCM

Device	1	2	3	4	5	6	7	8
1	0.90	0.20	0.18	0.08	0.60	0.15	0.06	0.02
2	0.07	0.62	0.69	0.86	0.31	0.72	0.19	0.06
3	0.03	0.18	0.13	0.06	0.09	0.13	0.75	0.92

TABLE V. CLUSTERING RESULT WITH FCM

Device	1	2	3	4	5	6	7	8
DC	1	2	2	2	1	2	3	3

Table VI shows the device similarity measured with Pearson correlation metric, which is used for DBSCAN clustering. And table VII shows which DC the devices belong to using DBSCAN clustering algorithm.

TABLE VI. SIMILARITIES BETWEEN DEVICES IN PEARSON CORRELATION COEFFICIENT

Device	1	2	3	4	5	6	7	8
1	1	0.94	0.90	0.48	0.91	0.89	0.85	0.92
2	0.94	1	0.94	0.48	0.93	0.93	0.86	0.91
3	0.90	0.94	1	0.43	0.95	0.93	0.90	0.88
4	0.48	0.48	0.43	1	0.45	0.39	0.44	0.47
5	0.91	0.93	0.95	0.45	1	0.89	0.92	0.93
6	0.89	0.93	0.93	0.39	0.89	1	0.86	0.84
7	0.85	0.86	0.90	0.44	0.92	0.86	1	0.88
8	0.92	0.91	0.88	0.47	0.93	0.84	0.88	1

TABLE VII. CLUSTERING RESULT WITH DBSCAN

Device	1	2	3	4	5	6	7	8
DC	1	1	1	2	1	1	3	3

It could be seen from these tables that the selection of different clustering algorithm only has slight influence on clustering results. Device 2, 3, 6 are always classed into the same cluster, so as the device 7 and 8, which confirms the intuition that different devices with the same type should be clustered into a same group because of having the nearest radio physical feature. Device 1, 4 and 5 show different properties when different clustering algorithms are applied.

Because of using the same distance measurement (Euclidean distance), both AP and FCM algorithms obtain very close clustering result except device 5. When the Pearson correlation metric is used, the device 4 demonstrates the greatest diversity. Thus the main influence factor of device clustering comes from distance measurement rather than clustering algorithm.

C. *Computational Cost*

The computing overhead of fingerprint pre-processing and tracking phase is the same. Here we only give the computational cost of the three clustering algorithms, as table VIII shows.

TABLE VIII. COMPUTATIONAL COMPLEXITY COMPARISON WITH AP/FCM/DBSCAN ALGORITHM

Clustering algorithm	AP	FCM	DBSCAN
Computational complexity	O(N*N*T)	O(N*C*T)	O(N*log(N))

In this table, N means the number of fingerprints, T is the number of iterations and C refers to the number of clusters. It could be found that none of these clustering algorithms owns most optimal computing overhead. The number of fingerprints, the iterations and even the selection of clusters all influence the final run time. Therefore there not exits the best algorithm in both clustering effect and computational complexity. The only standard is to depend on the practical situation.

D. *Localization Accuracy Analysis*

To test the localization performance with the above three device clustering algorithms (AP/FCM/DBSCAN) over diverse devices, we perform three combined localization experiments. The combined parameter is defined as table IX. The original mixing database is also used as a comparison.

Table IX shows the abbreviation.

TABLE IX. TERMS ABBREVIATION

Abbreviation	Description
C	Likelihood function obtained with clustered fingerprint data
M	Likelihood function obtained with all the

	original mixed fingerprint data
G	Using Likelihood of Gaussian model
L	Using linear transformation between clusters

Figure 3 compares the localization error with different combinations-CLG, MG and CG. The localization algorithm with CG represents the highest accuracy. And from the CLG and MG curve, we can see that the device heterogeneity has a large impact on the localization accuracy. The MG curve represents lowest accuracy. By performing the linear transformation, it maps the fingerprints of tracking phase to the fingerprints of training phase, which improves the localization accuracy remarkably.

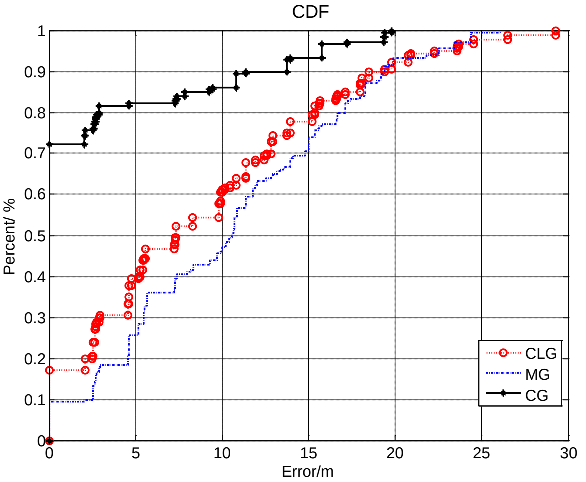

Figure 3. Localization error of different combination

The linear relations between every two clusters are shown in Figure 4 and the linear coefficient is embedded in the top half part of figure. From this figure, it can be seen that the linear relationship has maintained after various devices are classed into different DCs.

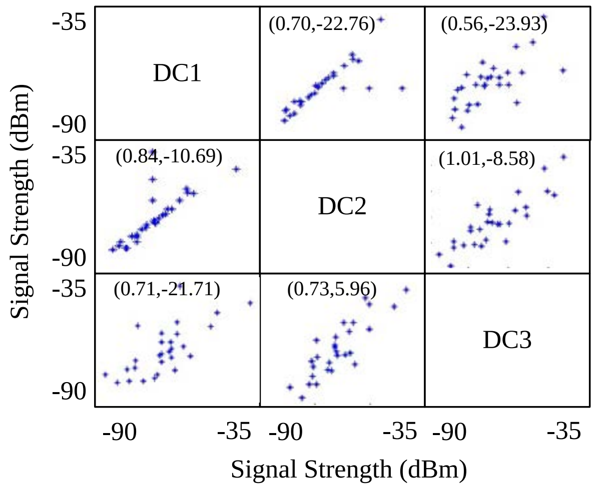

Figure 4. Linear relations between Device-Clusters

V. Conclusion and Future Work

In this paper, we compare the effect of three device clustering algorithms. Based on the macro Device-Cluster rather than original devices, the number of device types which the DC system maintains decreases greatly. Besides, the sampling overhead to each device in total space is decreased. More importantly, DC algorithm suits different clustering algorithms like FCM, AP and DBSCAN so that the specific selection of clustering algorithm depends on practical situation.

However, this experiment only employs 6 different types of mobile smart devices to cluster, which is not enough. In the future, we consider deploying the DC algorithm in complex metropolitan environment, ample volunteers and mobile devices to verify its performance. At the same time, how to reduce the computational overhead of training phase also merits future concern. Because in the real-time positioning environment, the time-consuming clustering will also affect the localization effect.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (60973110),the Natural Science Foundation of Beijing City of China (4102059),the Major Projects of Ministry of Industry and Information Technology (2010ZX03006-002-03, 2011ZX03005-005), the Guangdong Chinese Academy of Science Comprehensive Strategic Cooperation Project (2011A090100016), Jiangsu Provincial Science and Technology Support Program (BE2010017), the National Internet of Things Development Special Fund Project (2011-05) and Tianjin Binhai New Area Science Little Giant Enterprises Growth Plan (2011-XJR12009).

References

[1] Mooney, P. and Corcoran, P. Using OSM for LBS: An analysis of changes to attributes of spatial objects. 8th International Symposium on Location-Based Services (LBS 2011), Vienna, 21-23 November 2011.

[2] Nisarg Kothari, Balajee Kannan, Evan D. Glasgwow, M. Bernardine Dias: Robust Indoor Localization on a Commercial Smart Phone. Procedia CS 10: 1114-1120 (2012).

[3] Arvin Wen Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, Hao-Hua Chu, Accuracy Performance Analysis between War Driving and War Walking in Metropolitan WiFi Localization, IEEE Transactions on Mobile Computing, vol. 9, no.11, pp. 1551-1562, November 2010.

[4] V. Honkavirta and T. Perala, "A comparative survey of WLAN location fingerprinting methods," WPNC 2009 pp. 243-251.

[5] J. Park, D. Curtis, S.J. Teller, and J. Ledlie, "Implications of device diversity for organic localization", in Proc. INFOCOM, 2011, pp.3182-3190.

[6] Minkyu Lee, Hyunil Yang, Dongsoo Han, Chansu Yu: Crowdsourced radiomap for room-level place recognition in urban environment. PerCom Workshops 2010: 648-653.

[7] Arvin Wen Tsui, Yu-Hsiang Chuang, Hao-Hua Chu: Unsupervised Learning for Solving RSS Hardware Variance Problem in WiFi Localization. MONET 14(5): 677-691 (2009).

[8] Huang CHENG, Hai-yong LUO, Fang ZHAO, Device-clustering algorithm in crowdsourcing-based localization, The Journal of China Universities of Posts and Telecommunications, Volume 19, Supplement 2, October 2012, Pages 114–121.

[9] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical Robust Localization over Large-Scale 802.11 Wireless Networks," in Proc. Annual International Conference on

Mobile Computing and Networking (MobiCom), Philadelphia, PA, Sep. 2004, pp. 70–84.

[10] Mikkel Baun Kjærgaard: Indoor location fingerprinting with heterogeneous clients. Pervasive and Mobile Computing 7(1): 31-43 (2011)

[11] Kjærgaard, M.B., Munk, C.V.: Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength (concise contribution). In: Sixth Annual IEEE International Conference on Pervasive Computing and Communications, pp. 110–116 (2008).

[12] Fangfang Dong, Yiqiang Chen, Junfa Liu, Qiong Ning, Songmei Piao: A Calibration-Free Localization Solution for Handling Signal Strength Variance. MELT 2009: 79-90.

[13] Zheng, V.W., Pan, S.J., Yang, Q., Pan, J.J.: Transferring multi-device localization models using latent multi-task learning. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, pp. 1427–1432 (2008).

[14] Sun, Z., Yiqiang Chen, J.Q., Liu, J.: Adaptive localization through transfer learning in indoor wi-fi environment. In: Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications, pp. 331–336 (2008).

[15] Lloyd, S. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory 28 (2): 129–137. doi:10.1109/TIT.1982.1056489.

[16] Nock, R. and Nielsen, F. (2006) "On Weighting Clustering", IEEE Trans. on Pattern Analysis and Machine Intelligence, 28 (8), 1–13.

[17] Frey, B.J., Dueck, D. (2007). Clustering by Passing Messages Between Data Points. Science, 315(5814), 972-976. DOI: 10.1126/science.1136800.

[18] Sander, Jörg; Ester, Martin; Kriegel, Hans-Peter; Xu, Xiaowei (1998). "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications". Data Mining and Knowledge Discovery (Berlin: Springer-Verlag) 2 (2): 169–194. doi:10.1023/A:1009745219419.

[19] Xing-chuan Liu, Sheng Zhang, Qing-yuan Zhao, Xiao-kang Lin, "A real-time algorithm for fingerprint localization based on clustering and spatial diversity", Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on, On page(s): 74 – 81.

[20] Seshadri V, Zaruba GV, Huber M (2005) A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication, in Proceedings of IEEE International Conference on Pervasive Computing (PERCOM 2005), pp. 75–84.

[21] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanan. A probabilistic approach to WLAN user location estimation. International Journal of Wireless Information Networks, 9(3), July 2002.