

國立交通大學

網路工程研究所

碩士論文

在大型網路下以群簇法為
基礎的樣本比對定位法之研究



Cluster-Based Pattern-Matching Localization Schemes for
Large-Scale Wireless Networks

研究生：吳秉禎

指導教授：曾煜棋 教授

中華民國九十六年六月


在大型網路下以群簇法為基礎的樣本比對定位法之研究

學生: 吳秉禎

指導教授: 曾煜棋老師

國立交通大學資訊工程學系(研究所) 碩士班

摘要



在定位服務裡, 系統的反應時間是一個關鍵點, 對於即時性的應用來說, 更是如此。在大型網路下 (如無線城市), 以樣本比對法為基礎的定位系統, 如此的需求更為明顯。此類定位法的運作是仰賴目前物體收集到的訊號強度特徵與事先在訓練階段建立的以訊號強度為樣本的資料庫做比對來達到定位的目的。在這篇論文中, 我們提出一個以群簇法為基礎的樣本比對定位架構來加快定位的程序。藉著將擁有類似的訊號特徵樣本的訓練點群聚在一起, 我們會展示如何降低定位所需的比較複雜度來加速整個定位的流程。為了解決訊號飄移的問題, 我們更提出了幾個有效的分群法。在許多廣泛的模擬的結果下, 我們可以發現: 平均來說, 在不影響定位準確度的情況下, 我們提出的系統相較於原來的樣本比對法的比較複雜度上可減少至少 90%。

Cluster-Based Pattern-Matching Localization Schemes for Large-Scale Wireless Networks

Student: Bing-Jhen Wu

Advisor: Prof. Yu-Chee Tseng

Department of Computer Science and Information Engineering
National Chiao Tung University

Abstract

In location-based services, the response time of location determination is critical, especially in real-time applications. This is especially true for pattern-matching localization methods, which rely on comparing an object's current signal strength pattern against a pre-established location database of signal strength patterns collected at the training phase, when the sensing field is large (such as a wireless city). In this work, we propose a cluster-based localization framework to speed up the positioning process for pattern-matching localization schemes. Through grouping training locations with similar signal strength patterns, we show how to reduce the associated comparison cost so as to accelerate the pattern-matching process. To deal with signal fluctuations, several clustering strategies are proposed. Extensive simulation studies are conducted. Experimental results show that more than 90% computation cost can be reduced in average without degrading the positioning accuracy.

誌謝

這篇論文的完成,首先要感謝曾教授所給予的指導與意見,不僅僅提升整體論文的品質更帶領我了解到整個研究的過程。感謝在過程中協助良多的郭聖博學長,對於整篇論文的所提供的寶貴的建議與許多的幫助,讓此篇論文得以順利完成。也感謝實驗室夥伴們彼此間的加油打氣,讓我在研究的路上不孤單。最後,感謝我的家人兩年來一直在背後默默地支持,讓我能專心在自己的研究上。



Contents

中文摘要	i
Abstract	ii
誌謝	iii
Contents	iv
List of Figures	1
1 Introduction	2
2 Related Works	5
3 The Cluster-Based Pattern-Matching Localization Framework	8
3.1 The Training Phase	8
3.2 The Positioning Phase	10
4 Clustering Algorithms	11
4.1 k -means Algorithm	11

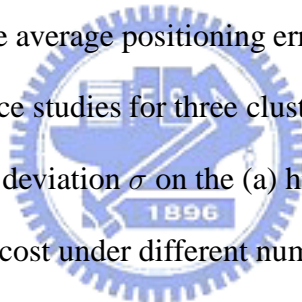


4.2	Clustering Techniques Allowing Overlaps	13
4.2.1	Multi-Nearest-Neighbor Strategy	13
4.2.2	Voronoi-based Strategy	14
4.2.3	Probability-based Strategy	16
5	Simulations	20
5.1	Simulation Model	20
5.2	Impact of Clustering on the Average Positioning Error	22
5.3	Sensitive Performance Study for Clustering Strategies	23
5.4	Performance Comparison of Clustering Strategies	25
5.5	Performance Study of Total Comparison Cost	25
6	Conclusion	27
	Bibliography	28



List of Figures

3.1	The cluster-based localization framework.	9
4.1	An example of the problem of k -means algorithm.	12
4.2	An example of Voronoi-based Overlapping mechanism.	15
5.1	The comparison of the average positioning error under different σ	22
5.2	Sensitivity performance studies for three clustering strategies.	24
5.3	Effect of the standard deviation σ on the (a) hit rate and (b) average cluster size.	26
5.4	The total comparison cost under different number of clusters.	26



Chapter 1

Introduction

Location-based services (LBSs) have emerged as one of the killer applications for mobile computing and wireless data services. While providing great market values to business applications, such services are also critical to public safety, transportation, emergency response, and disaster management. Consequently, location estimation is essential to the success of LBSs. In addition to the well-known GPS [1], a lot of techniques have been proposed for indoor localization, such as infrared-based [2], ultrasonic-based [3], and RF-based [4, 5] systems.

Among all localization systems, the RF-based systems are probably most cost-effective because they can rely on existing wireless network infrastructures (such as IEEE 802.11 WLAN). However, such systems need to handle the characteristic of signal strengths, which may fluctuate frequently. The *pattern-matching* schemes [4, 5, 6, 7, 8], or known as the *fingerprinting* schemes, deal with this problem by involving two phases: *training* and *positioning*. In the training phase, given a set of training locations, the received signal strengths of all base stations (or beacons) at these locations are collected for a sufficient amount of time. Therefore, for each training location, a feature vector is calculated. Then, in the positioning phase, when an object needs to determine its location, it can compare its current received signal strengths against the

feature vectors in the location database to check their similarity. The corresponding location of the most similar feature vector is selected as the possible location of the object.

Recently, many literatures apply pattern-matching localization methods to a large-scale environment [9, 10, 11, 12]. However, they may encounter the scalability problem because of the huge calibration efforts required in the training phase and high comparison cost spent in the positioning phase. For example, in a wireless city, thousands or millions of training records may have to be collected in a location database. Several efforts have been dedicated to reducing the calibration cost [10, 13, 14, 15]. In this paper, we aim to reduce the computation cost incurred in the positioning phase. This would enable us to support real-time LBSs. We propose a cluster-based localization framework which also consists of two phases. In the training phase, similar to existing pattern-matching approaches, we first collect feature vectors of training locations. Through clustering techniques, those training locations with similar feature vectors are grouped together. This results in a small number of clusters. For each cluster, a representative feature vector is derived. Then in the positioning phase, given a signal strength vector, we first compare it against all clusters' representative feature vectors and pick the one whose representative feature vector is most similar to the given signal strength vector. Finally, only the training locations in the selected cluster are further evaluated to determine the estimated location of the object.

Although the clustering technique is able to reduce the computation cost, its positioning accuracy may be reduced if the right cluster is not selected. If a false cluster is selected, the final location estimation may be incorrect. We refer to this as the *false cluster selection*. Apparently, the probability of a false cluster selection should be reduced. In this paper, we propose several clustering strategies. First, we show that the traditional k -means algorithm [16] is not suitable when the effect of noise is not negligible. Then we propose three clustering strategies to enhance the k -means algorithm by allowing clusters to have overlapping members. Although having

duplications is redundant, it can effectively reduce the events of false cluster selection due to noises. To verify our results, a simulation model is built and extensive simulation studies are conducted. Experimental results show that this framework is able to reduce at least 90% computation cost without sacrificing accuracy.

The rest of this paper is organized as follows. Chapter 2 discusses some reviews. The proposed cluster-based framework is described in Chapter 3. Chapter 4 presents several clustering strategies. Chapter 5 contains our performance studies. Chapter 6 concludes this paper.



Chapter 2

Related Works

Several localization systems have explored the pattern-matching techniques. In [4], the nearest-neighbor algorithm is applied to search the location database for the training location with the shortest Euclidean distance in the signal space. Based on probability theory, [5] presents a probabilistic framework for localization to handle signal strength fluctuations. Reference [8] adopts the similar concept to develop recursive Bayesian filters for localization. In general, the nearest-neighbor approach is not as effective as the probabilistic one [17].

In [6], a more sophisticated network-based classification method is proposed. A neural network, which consists of multiple layers of interconnected neurons is adopted to model the dependencies among a set of random variables. It has a forward and back propagation mechanism to adaptively assign suitable weights to neuron connections in the training phase. Then, the well trained network can be used to classify an observed sample of signal strengths in the positioning phase. Based on the statistical learning theory, [7] proposes a *support vector machine (SVM)* to find a high-dimensional hyperplane such that any two training data set can be partitioned between two sides of this plane and their distances to this plane can be as far as possible.

For large-scale environments, some literatures have considered the scalability issue incurred in the training phase [13, 14, 11] and the positioning phase [18, 19]. In the training phase, to relieve huge labor cost needed for training data collection, an intuitive idea is to collect less training locations. However, it also represents that we cannot capture the detailed signal strength patterns so the positioning results will be coarse-grained. Hence, [13] proposes to generate a small number of virtual training locations from the actual ones by interpolation techniques. Similarly, multidimensional regression [14] is used to build a nonlinear mapping between the signal space and the physical space. For the reason that manually collecting training samples with the correct location labels is time-consuming, [13] suggests to use unlabeled user traces to compensate the loss of accuracy caused by a relatively small number of training locations. An unlabeled user trace is a sequences of continuously received signal strength measurements without location labels. With the help of a hidden Markov model to model user traces, unlabeled user traces can be used to simplify the training process while keeping a certain degree of positioning accuracy. Furthermore, a calibration-free mechanism is the extreme solution to save labor cost. Without a training phase, signal propagational models can be used to predict the characteristics of signal strengths in the environment [4]. However, such systems will have higher positioning error because multipath fading and interference are hard to be precisely modeled in an indoor environment.

The issue of reducing the real-time comparison cost in the positioning phase is discussed in [18, 19]. Their main ideas are both to apply clustering techniques to the training locations, so only a subset of them needs to be searched. Reference [18] constructs clusters according to the physical coordinates of training locations. It claims that the estimated locations of two consecutive location queries should be very close in the physical space. Thus, only the training locations close to the previous estimated one need to be searched for the current query. However,

it does not consider the actual signal space and its searching range strongly relies on the query interval and the user mobility model.

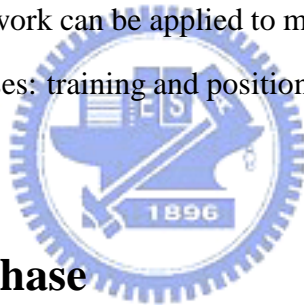
In [19], training locations that see the q strongest signal strengths from the same q access points (APs) are grouped together. This clustering technique is simple but has several drawbacks. First, the top q APs with the strongest signals at a fixed location may vary over time, thus causing false cluster selection. Second, the number of clusters is not a controllable parameter. In our work, the number of clusters is tunable and false cluster selection can be effectively avoided.



Chapter 3

The Cluster-Based Pattern-Matching Localization Framework

The proposed clustering framework can be applied to most pattern-matching localization methods. It also consists of two phases: training and positioning. Fig. 3.1 depicts the structure of the framework.



3.1 The Training Phase

We assume that there are m beacons (or APs), denoted as $\mathcal{B} = \{b_1, b_2, \dots, b_m\}$, being deployed in the field. In this field, we define n training locations $\mathcal{L} = \{\ell_1, \ell_2, \dots, \ell_n\}$. Let the feature space $\mathcal{F} \in \mathbb{R}^m$, where \mathbb{R} is the set of possible signal strengths. For each training location ℓ_i , $i = 1..n$, we collect a sufficient number of training samples from beacons and calculate the *feature vector* $\mathbf{v}_i = [v_{i,1}, v_{i,2}, \dots, v_{i,m}] \in \mathcal{F}$ for ℓ_i , where $v_{i,j}$ is the average received signal strength from b_j at ℓ_i . For those training locations with similar feature vectors, we exploit clustering techniques to group them together. Specifically, we will compute k location sets $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$ such that $\mathcal{C}_i \subseteq \mathcal{L}$, $i = 1..k$, and $\bigcup_{i=1}^k \mathcal{C}_i = \mathcal{L}$. The detail clustering algorithms

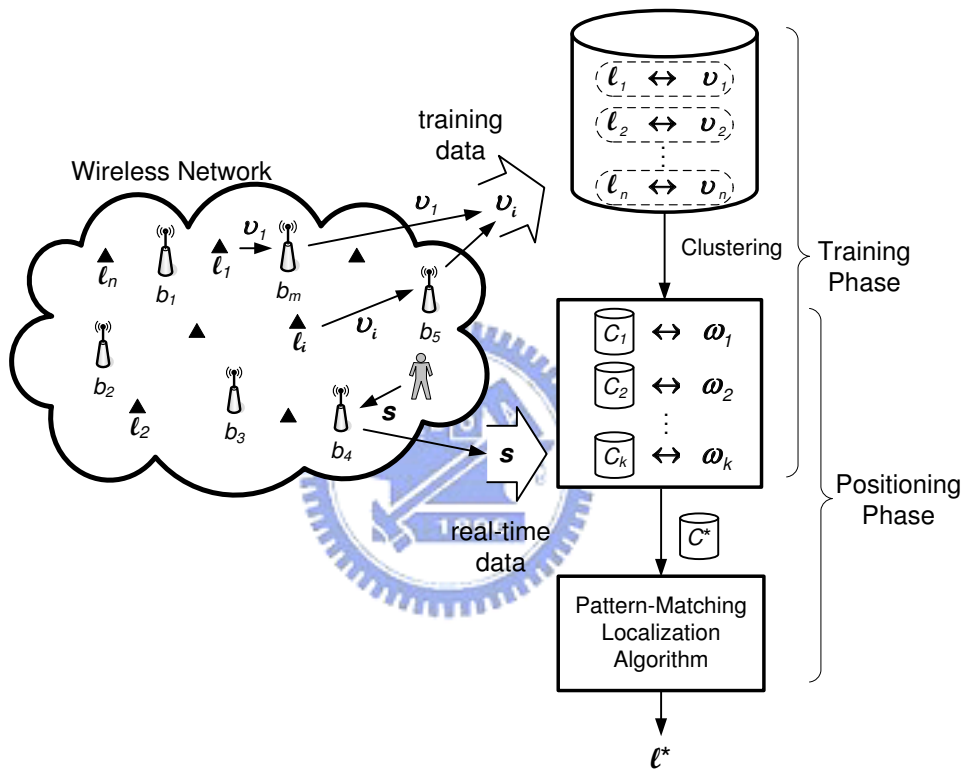


Figure 3.1: The cluster-based localization framework.

will be given in Section 4. For each location set \mathcal{C}_i , its representative feature vector is expressed by $\boldsymbol{\omega}_i = [\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,m}] \in \mathcal{F}$, where $\omega_{i,j}$ is the average of signal strengths $\frac{\sum_{x:\ell_x \in \mathcal{C}_i} v_{x,j}}{|\mathcal{C}_i|}$. Note that two location sets may have overlaps, i.e., $\mathcal{C}_i \cap \mathcal{C}_j$ is not necessarily an empty set.

3.2 The Positioning Phase

In the positioning phase, when an object needs to determine its location, we can measure its signal strength vector $\boldsymbol{s} = [s_1, s_2, \dots, s_m]$, where s_j is the signal strength of b_j . Our goal is to determine the object's location in a real-time manner. In typical pattern-matching methods, \boldsymbol{s} will be compared to all n feature vectors in the location database. However, in a large-scale field (such as a wireless city), thousands or millions of vectors may need to be compared. By clustering training locations with similar feature vectors into a group, we only need to compare \boldsymbol{s} against the representative feature vector $\boldsymbol{\omega}_i$ of each \mathcal{C}_i first. As in most works, the similarity between \boldsymbol{s} and \mathcal{C}_i is defined as the Euclidean distance of their feature vectors in \mathcal{F} , $\text{sim}(\boldsymbol{s}, \mathcal{C}_i) = \|\boldsymbol{s}, \boldsymbol{\omega}_i\| = \sqrt{\sum_{j=1}^m (s_j - \omega_{i,j})^2}$. Then, the most similar cluster, denoted by \mathcal{C}^* , is selected, i.e., $\mathcal{C}^* = \arg \min_{\mathcal{C}_i} \text{sim}(\boldsymbol{s}, \mathcal{C}_i)$. That is, only the training locations in \mathcal{C}^* will be further searched. We refer to this as the Nearest Neighbor in Signal Space (NNSS) algorithm [4]. In NNSS, users' locations are estimated by comparing \boldsymbol{s} against each training location ℓ_i in \mathcal{C}^* according to the Euclidean distance, i.e., $\text{sim}(\boldsymbol{s}, \ell_i) = \|\boldsymbol{s}, \boldsymbol{v}_i\| = \sqrt{\sum_{j=1}^m (s_j - v_{i,j})^2}$ in \mathcal{F} . The estimated location is $\ell^* = \arg \min_{\ell_i \in \mathcal{C}^*} \text{sim}(\boldsymbol{s}, \ell_i)$. Therefore, the computation cost is decreased from $O(|\mathcal{L}|)$ to $O(k + \frac{|\mathcal{L}|}{k})$ if any two location sets are disjoint.

Chapter 4

Clustering Algorithms

Below, we propose several clustering algorithms to partition the location database. We start with the well-known k -means algorithm, followed by three enhanced clustering strategies.

4.1 k -means Algorithm



The k -means algorithm developed in [16] can be applied to our model. There are multiple iterations. In the x -th iteration, we will form k clusters $\mathcal{C}_1^{(x)}, \mathcal{C}_2^{(x)}, \dots, \mathcal{C}_k^{(x)}$. Initially, we construct k seeds $\omega_1^{(0)}, \omega_2^{(0)}, \dots, \omega_k^{(0)} \in \mathcal{F}$, where each seed $\omega_i^{(0)}$, $i = 1..k$, is randomly selected from the set of feature vectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, and $\omega_i^{(0)} \neq \omega_j^{(0)}$ for all $i \neq j$. (Other ways to choose the initial values of $\omega_1^{(0)}, \omega_2^{(0)}, \dots, \omega_k^{(0)}$ are discussed in [20]. Here we adopt the random strategy.) With these seeds, we define cluster $\mathcal{C}_i^{(1)}$ in the first iteration as follows:

$$\ell_j \in \mathcal{C}_i^{(1)} \Leftrightarrow \omega_i^{(0)} = \arg \min_{\omega_y^{(0)}} \|\mathbf{v}_j, \omega_y^{(0)}\|.$$

That is, ℓ_j will be categorized as a member of cluster $\mathcal{C}_i^{(1)}$ if \mathbf{v}_j is closest to the seed $\omega_i^{(0)}$ among all other seeds. From each cluster $\mathcal{C}_i^{(1)}$, $i = 1..k$, we then calculate a new seed $\omega_i^{(1)}$ by averaging

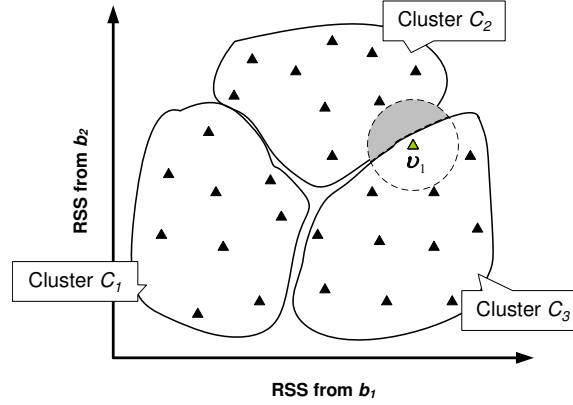


Figure 4.1: An example of the problem of k -means algorithm.

the received signal strengths for all $\ell_j \in \mathcal{C}_i^{(1)}$,

$$\omega_i^{(1)} = \text{avg}\{\mathbf{v}_j : \forall \ell_j \in \mathcal{C}_i^{(1)}\}.$$

In the x -th iteration, for all $x \geq 2$, according to the seeds generated in the $(x - 1)$ -th iteration, we define cluster $\mathcal{C}_i^{(x)}$ as follows:

$$\ell_j \in \mathcal{C}_i^{(x)} \Leftrightarrow \omega_i^{(x-1)} = \arg \min_{\omega_y^{(x-1)}} \|\mathbf{v}_j, \omega_y^{(x-1)}\|.$$

Similarly, from each cluster $\mathcal{C}_i^{(x)}$, we can calculate another seed $\omega_i^{(x)} = \text{avg}\{\mathbf{v}_j : \forall \ell_j \in \mathcal{C}_i^{(x)}\}$. The regrouping processes will be repeated iteratively until the condition $\mathcal{C}_j^{(x)} = \mathcal{C}_j^{(x+1)}$ is satisfied for all $j = 1..k$. At last, we obtain $\omega_j = \omega_j^{(x)}$ for each location set $\mathcal{C}_j = \mathcal{C}_j^{(x)}$.

Ideally, in the positioning phase, when an object provides its current signal strength vector \mathbf{s} , we would expect that a correct cluster with the most similar feature vector can be selected. However, due to the fluctuation of radio signal, this cannot always be achieved. Fig. 4.1 shows an example with three clusters in a feature space. Due to signal fluctuation, the signal strength vector \mathbf{s} of an object which locates at ℓ_1 may appear in multiple clusters. As shown by dotted circles in the figure, the distribution of \mathbf{s} is modeled by an uniform distribution for ease of

discussion. According to the k -means algorithm, if s is in the gray region, it is more similar to the cluster \mathcal{C}_2 than \mathcal{C}_1 and \mathcal{C}_3 . Hence, a false cluster selection happens. Clearly, this situation is more serious near the boundary of clusters.

4.2 Clustering Techniques Allowing Overlaps

The problem shown in Fig. 4.1 calls for the design of clusters with certain degrees of overlaps. Below, we propose three clustering schemes extended from the k -means algorithm, which allow a training location to join multiple clusters. We define *overlapping degree* λ to be the average number of clusters that training locations can join. Clearly, this will increase the searching complexity in the positioning phase to $O(k + \lambda \times \frac{|\mathcal{L}|}{k})$.

All schemes are similar to the k -means algorithm partitioning \mathcal{L} into k sets. The main difference is the way to determine which clusters a training location will join. In the first *multi-nearest-neighbor* strategy, each training location can join the first few clusters closest to it. In the second *Voronoi-based* strategy, the overlapping degree is determined by the geometric characteristic of the distribution of clusters. The last *probability-based* strategy can adaptively adjust the overlapping degree of each training location according to the levels of environmental noise.

4.2.1 Multi-Nearest-Neighbor Strategy

The multi-nearest-neighbor strategy assigns a constant overlapping degree λ_M to all training locations. As mentioned before, k clusters of training locations $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$ are obtained by the k -means clustering algorithm. Then, for each training location ℓ_i , it will join the top λ_M similar clusters, which are ranked by the inverse of Euclidean distance $1/\|\mathbf{v}_i, \boldsymbol{\omega}_j\|$. Averagely,

the searching space is increased from $O(k + \frac{|\mathcal{L}|}{k})$ to $O(k + \lambda_M \times \frac{|\mathcal{L}|}{k})$ compared to the k -means clustering algorithm.

This strategy allows each training location to join multiple closest clusters in the feature space, unlike the single one in the k -means clustering algorithm. It is an intuitive solution for solving the signal fluctuation problem. If samples of a location are possible to be estimated to many nearby clusters, there is no reason to make this training location join only one cluster. For the example illustrated in Section 4.1, we can avoid incorrect location estimations caused by false cluster selection if ℓ_1 is allowed to join two closest clusters \mathcal{C}_2 and \mathcal{C}_3 simultaneously.

4.2.2 Voronoi-based Strategy

Although the multi-nearest-neighbor strategy is simple to be implemented and easy to control the average searching space, the parameter λ_M is hard to determine. If λ_M is too small, it may not compensate for the effect of signal fluctuations. Thus, the problem of false cluster selection remains. On the other hand, if λ_M is too large, some training locations may join unnecessary clusters, thus causing redundancy. We can observe that when \mathbf{v}_i of a training location is close to the center of a cluster in \mathcal{F} , the number of clusters it joins should not be the same as another training location whose feature vector \mathbf{v}_j is near the periphery of a cluster.

For this consideration, we next propose the Voronoi-based strategy. After performing the k -means algorithm, $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ is decomposed into k partitions centered at ω_j , $1 \leq j \leq k$. It can be observed that $\|\mathbf{v}_i, \omega_x\| \leq \|\mathbf{v}_i, \omega_y\|$ for all $y \neq x$ if $\ell_i \in \mathcal{C}_x$. This property is equivalent to a Voronoi diagram [21], where all points in a Voronoi cell are closest to the Voronoi vertex in the same cell. Thus, the members of a cluster \mathcal{C}_x are contained in a Voronoi cell V_x with a Voronoi vertex at ω_x . If we let a training location which is close to Voronoi edges join more

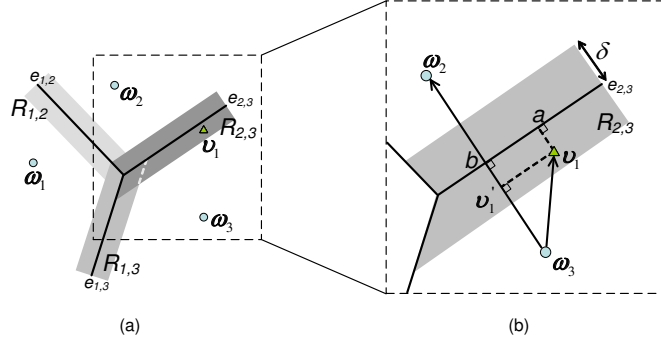


Figure 4.2: An example of Voronoi-based Overlapping mechanism.

clusters and oppositely let the others join less clusters, we can improve the effectiveness of the overlapping technique.

Motivated by the observation above, we propose the Voronoi-based strategy. For each neighboring Voronoi cells V_x and V_y , we formally define an overlapping region $R_{x,y}$ ($x < y$) in which any training location whose feature vector located joins both \mathcal{C}_x and \mathcal{C}_y . For example, in Fig. 4.2(a), there are three Voronoi cells V_1 , V_2 , and V_3 , separated by three Voronoi edges $e_{1,2}$, $e_{2,3}$, and $e_{1,3}$, and three overlapping regions $R_{1,2}$, $R_{1,3}$, $R_{2,3}$, are shaded. The feature vector v_1 is inside $R_{2,3}$ and v_2 is located both in $R_{2,3}$ and $R_{1,3}$. As a result, ℓ_1 joins \mathcal{C}_2 and \mathcal{C}_3 ; while ℓ_2 joins \mathcal{C}_1 , \mathcal{C}_2 , and \mathcal{C}_3 . An overlapping region $R_{x,y}$ can be regarded as an expansion of an Voronoi edge $e_{x,y}$ along the edges incident to the endpoints of $e_{x,y}$, like the gray region $R_{2,3}$ shown in Fig. 4.2(b). The expansion range δ is used to control the size of $R_{x,y}$ by expanding both sides from $e_{x,y}$.

To determine which overlapping regions where a feature vector v_i located, we have to determine the Voronoi cell V_x such that $\ell_i \in \mathcal{C}_x$ and a neighboring Voronoi cell V_y of V_x . Let $dist(v_i, e_{x,y})$ be the vertical distance between v_i and the Voronoi edge $e_{x,y}$. Then, if $dist(v_i, e_{x,y}) < \delta$, then v_i is definitely in $R_{x,y}$. Hence, ℓ_i will join \mathcal{C}_y in the Voronoi-based strategy.

However, due to the costly computation of $e_{x,y}$ in high dimension feature space [21], we do not calculate $dist(\mathbf{v}_i, e_{x,y})$ directly. Instead, we use the projection of the vector $\overline{\omega_x \mathbf{v}_i}$ on the line $\overline{\omega_x \omega_y}$ to obtain $dist(\mathbf{v}_i, e_{x,y})$. Again in Fig. 4.2(b), we want to determine $dist(\mathbf{v}_1, e_{2,3}) = |\overline{\mathbf{v}_1 a}|$. First, $\overline{\omega_3 \mathbf{v}_1}$ is projected on $\overline{\omega_3 \omega_2}$ as $\overline{\omega_3 \mathbf{v}'_1}$. Let b be an intersection point of $\overline{\omega_3 \omega_2}$ and $e_{2,3}$. The edge $e_{2,3}$ and $\overline{\omega_3 \omega_2}$ are mutually orthogonal, which is a property of a Voronoi diagram. Therefore, the points \mathbf{v}_1 , \mathbf{v}'_1 , b , and a form a rectangle so $|\overline{\mathbf{v}'_1 b}| = |\overline{\mathbf{v}_1 a}|$. Finally, $|\overline{\mathbf{v}'_1 b}|$ can be obtained by $|\overline{\omega_3 b}| - |\overline{\omega_3 \mathbf{v}'_1}| = |\overline{\omega_2 \omega_3}|/2 - \overline{\omega_3 \mathbf{v}_1} \cdot \overline{\omega_3 \omega_2} / \|\overline{\omega_3 \omega_2}\|$. Compared with finding $|\overline{\mathbf{v}_1 a}|$ directly, this method saves more computation cost.

The above procedure functions well based on the assumption that each Voronoi cell knows the neighborhood information. For example, V_3 knows V_1 and V_2 are its neighbors in Fig. 4.2(b). Unfortunately, we cannot obtain this information until the relationship between Voronoi cells is completely discovered. This is as hard as finding Voronoi edges. Note that the k -means clustering algorithm only finds out the Voronoi vertex of each cell. Here, we propose a simple speculation technique, called *neighborhood speculation*, to guess the neighborhood relationship. It is based on an observation that if two Voronoi cells V_x and V_y are neighbors, then the midpoint of $\overline{\omega_x \omega_y}$ is *usually* closer to V_x and V_y than any other cell. Therefore, we use the position of the midpoint of V_x and V_y to speculate the relationship between them. If the midpoint is inside other cells except for V_x or V_y , we tend to believe that V_x and V_y are not neighbors.

4.2.3 Probability-based Strategy

So far, the above overlapping strategies cannot adaptively adjust the overlapping degree of each training location according to different levels of environmental noise. Besides, the proposed strategies are lack of guaranteeing the probability of correct cluster selection. Hence, we propose the probability-based strategy which can overcome these problems by an off-line analysis.

As we have mentioned, the received samples are uncertain because of signal fluctuations. This uncertainty is usually modeled by a zero-mean Gaussian normal distribution. Hence, we denote the possible received samples at ℓ_i as a vector of random variables $\mathbf{S}_i = [r_{i,1} + N_1, r_{i,2} + N_2, \dots, r_{i,m} + N_m]$, where $r_{i,j}$ is the expected signal strength of b_j at ℓ_i without fluctuations and $N_j = N(0, \sigma_j)$, $j = 1..m$, are independent and identically distributed zero-mean normal random variables with variances σ_j , $j = 1..m$.

Then, we define a random variable $Z_{x,y} = X - Y$, where X is the square of the Euclidean distance between a random sample \mathbf{S}_i collected at a fixed location ℓ_i and a cluster feature vector ω_x , and Y is the square of the Euclidean distance between \mathbf{S}_i and another ω_y . Then we have

$$\begin{aligned}
Z_{x,y} &= X - Y \\
&= \|\mathbf{S}_i, \omega_x\|^2 - \|\mathbf{S}_i, \omega_y\|^2 \\
&= \sum_{j=1}^m [(r_{i,j} + N_j) - \omega_{x,j}]^2 - [(r_{i,j} + N_j) - \omega_{y,j}]^2 \\
&= \sum_{j=1}^m 2(\omega_{y,j} - \omega_{x,j})(r_{i,j} + N_j) - (\omega_{y,j}^2 - \omega_{x,j}^2). \tag{4.1}
\end{aligned}$$

Assume the number of training samples is large, so we can expect $v_{i,j} = r_{i,j}$. Let $\Theta_j = \omega_{y,j}^2 - \omega_{x,j}^2$ and $\Phi_j = \omega_{y,j} - \omega_{x,j}$. Hence,

$$Z_{x,y} = \sum_{j=1}^m 2\Phi_j N_j + \sum_{j=1}^m (2\Phi_j v_{i,j} - \Theta_j). \tag{4.2}$$

Because all $N_j = N(0, \sigma_j)$, for $j = 1..m$, are *i.i.d.* and Φ_j , Θ_j , and $v_{i,j}$ are constants, $Z_{x,y}$ is still a normal distributed random variable. Its mean and variance is

$$\begin{aligned}
\mu_{x,y} &= \sum_{j=1}^m (2\Phi_j v_{i,j} - \Theta_j), \\
\sigma_{x,y}^2 &= \sum_{j=1}^m (2\Phi_j \sigma_j)^2.
\end{aligned}$$

Therefore, the probability of the event $X < Y$ is

$$\begin{aligned} & Pr(Z_{x,y} < 0) \\ &= \int_{-\infty}^0 \frac{1}{\sqrt{2\pi\sigma_{x,y}^2}} \exp\left(\frac{-(z_{x,y} - \mu_{x,y})^2}{2\sigma_{x,y}^2}\right) dz_{x,y}. \end{aligned} \quad (4.3)$$

Let $X < Y$ be equivalent to $X \leq Y$. For a randomly collected sample \mathbf{S}_i , if $\|\mathbf{S}_i, \boldsymbol{\omega}_x\|$ is smaller than $\|\mathbf{S}_i, \boldsymbol{\omega}_j\|$ for all $j = 1..k, j \neq x$, the estimated cluster will be \mathcal{C}_x . We can express the probability of this event by

$$\begin{aligned} & Pr(\mathcal{C}^* = \mathcal{C}_x) \\ &= Pr(Z_{x,1} \leq 0, Z_{x,2} \leq 0, \dots, Z_{x,x-1} \leq 0, \\ & \quad Z_{x,x+1} \leq 0, \dots, Z_{x,k} \leq 0). \end{aligned} \quad (4.4)$$

For ease of computation, we assume events $Z_{x,j} \leq 0$ for all $j = 1..k, j \neq x$, are independent. Thus, Eq. (4.4) can be rewritten as

$$Pr(\mathcal{C}^* = \mathcal{C}_x) = \prod_{\substack{j=1 \\ j \neq x}}^k Pr(Z_{x,j} \leq 0). \quad (4.5)$$

In the multi-nearest-neighbor strategy, ℓ_i is allowed to join the top λ_M close clusters in \mathcal{F} . Instead, in this strategy, ℓ_i can join different number of clusters based on Eq. (4.5). A probability threshold ξ is defined here to denote the expected probability of a correct cluster selection. Then, we sort the clusters according to $Pr(\mathcal{C}^* = \mathcal{C}_x)$ for all $x = 1..k$ in descending order. By this sequence, ℓ_i will join the clusters one by one until $\sum_x Pr(\mathcal{C}^* = \mathcal{C}_x) \geq \xi$.

In summary, this strategy provides a more effective and efficient way to determine the overlapping degree of each training location. There are two key advantages. First, it assigns the

overlapping degree of a training location by its possibility of correct cluster selection. Second, no matter how the environment changes, it assures that the clusters to which a training location belongs can cover most possible regions where its signal strengths would fluctuate in \mathcal{F} .

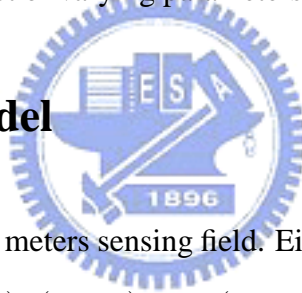


Chapter 5

Simulations

In this section, we conduct some experiments to evaluate the performance of our proposed framework. We study the impact of varying parameters used in our framework.

5.1 Simulation Model



We consider a 100×100 square meters sensing field. Eight beacons are placed at $(0, 0)$, $(0, 99)$, $(99, 0)$, $(0, 50)$, $(50, 0)$, $(50, 99)$, $(99, 50)$, and $(99, 99)$, respectively. As to other 9992 grid points, we collect 200 training samples at each of them in the training phase. The *log-distance path loss model* is exploited to model the signal propagation given by [22]:

$$PL(d) = PL(d_0) + 10\alpha \log\left(\frac{d}{d_0}\right) + N(0, \sigma), \quad (5.1)$$

where $d_0 = 1$ is the reference distance, and d is the distance between the transmitter and the receiver. α denotes the path loss exponent, typically from 2 to 6, and $N(0, \sigma)$ is a zero-mean normal distributed random variable with a standard deviation σ . Also, the transmit power P_t is set to be 15 dBm, $PL(d_0) = 37.3$, $\alpha = 2$, and $\sigma = 4$.

To evaluate the system performance, three performance metrics are employed:

- *Positioning error*: The error distance between the estimated location and the true location is the positioning error. We will use this metric to evaluate our proposal framework with other fingerprinted-based methods.
- *Hit rate*: To get insight into the impact of clustering on localization, the hit rate is defined as the probability of accurately predicting the cluster containing the true location. Obviously, the higher the hit rate, the less the positioning error caused by false cluster selection.
- *Average cluster size*: This metric stands for the improvement on computation reduction. According to our cluster-enhanced localization framework, the total number of comparisons would be $O(k + \lambda \times |\mathcal{L}|/k)$. Since the number of clusters is a tunable parameter in our proposed clustering strategies, we more care about the average number of training locations in clusters (i.e., $\sum_{i=1}^k |\mathcal{C}_i|/k$).

We evaluate the following clustering techniques: k -means algorithm, the Joint Clustering (abbreviated as *JC*) technique in [19], the multi-nearest-neighbor (abbreviated as *MNN*) strategy, the Voronoi-based (abbreviated as *Voronoi*) strategy, and the probability-based (abbreviated as *Prob*) strategy. A good clustering strategy should have a higher hit rate, a smaller average cluster size, and a lower positioning error.

The signal propagation model mentioned in Eq. (5.1) is used again in the positioning phase to simulate test samples. For each grid point, 200 test samples are generated and then a clustering technique is applied for each sample to determine its nearest cluster. A hit event occurs when the selected cluster contains the corresponding location of this sample.

To implement the probability-based strategy, we have to calculate the integrals in Eq. (4.5). However, it is not efficient and hard to obtain a precise product. Hence, in our simulations, we

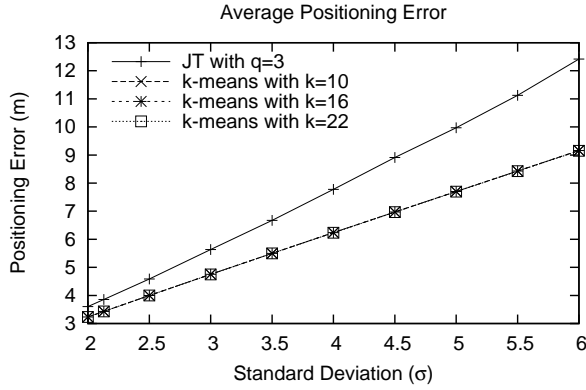


Figure 5.1: The comparison of the average positioning error under different σ .

approximate $Pr(C^* = C_x)$ by randomly generating h samples for each grid point ℓ_i according to the path loss model. After performing the probability-based clustering strategy to all these h samples, we obtain $Pr(C^* = C_x) = h_x/h$, where h_x is the number of samples whose closest cluster is C_x . In our simulation model, h is set to 1000.

5.2 Impact of Clustering on the Average Positioning Error

We first investigate the impact of clustering on the average position error. To demonstrate that clustering will also guarantee the accuracy of location estimation, we only compare the baseline clustering method (i.e. k -means algorithm) with the JC algorithm. We vary the standard deviation σ in the path loss model and show the effectiveness of clustering. Note that JC only generates 14 \sim 18 clusters in our simulation and thus we set the number of clusters for k -means algorithms to 10, 16 and 22, respectively. Fig. 5.1 shows the experimental results. In Fig. 5.1, JC incurs larger average positioning error under different noise levels. On the other hand, our proposed framework is able to provide better performance than that of JC in terms of average positioning error.

5.3 Sensitive Performance Study for Clustering Strategies

From the above experiment, our proposed framework with the baseline clustering algorithm (i.e., k -mean algorithm) outperforms the existing algorithm (i.e., JC). Before comparing k -means with other proposed strategies, we first conduct sensitive performance study for clustering strategies so as to determine the optimal parameter for each one. The number of clusters for each clustering strategy is set to 50, 100 and 200, respectively. Fig. 5.2 shows the performance study of the three clustering strategies with their parameters varied (λ_M in MNN, δ in Voronoi, and ξ in Prob). Note that to show their difference, we will compare these clustering strategies in terms of the hit rate and the average cluster size. It can be seen in Fig. 5.2(a) and Fig. 5.2(c), the hit rates of MNN and Voronoi have similar trend. However, Fig. 5.2(b) and Fig. 5.2(d) reveal that the Voronoi strategy can have smaller average cluster size in a lower density environment ($k = 50$). In other words, Voronoi is more suitable for a sparse environment. This agrees with our claim that Voronoi can effectively avoid training locations joining unnecessary clusters.

The performance study of the Prob strategy under different ξ is shown in Fig. 5.2(e) and Fig. 5.2(f). By comparing with other strategies, we have the following observations. First, both Voronoi and Prob have almost the same hit rate and their average cluster sizes are similar. Second, the hit rate of Prob is always above the required threshold ξ under different scenarios (i.e., different setting for k), which indicates that the Prob strategy can automatically adjust itself to adapt to different environments. As a result, the Prob strategy is superior than the Voronoi strategy.

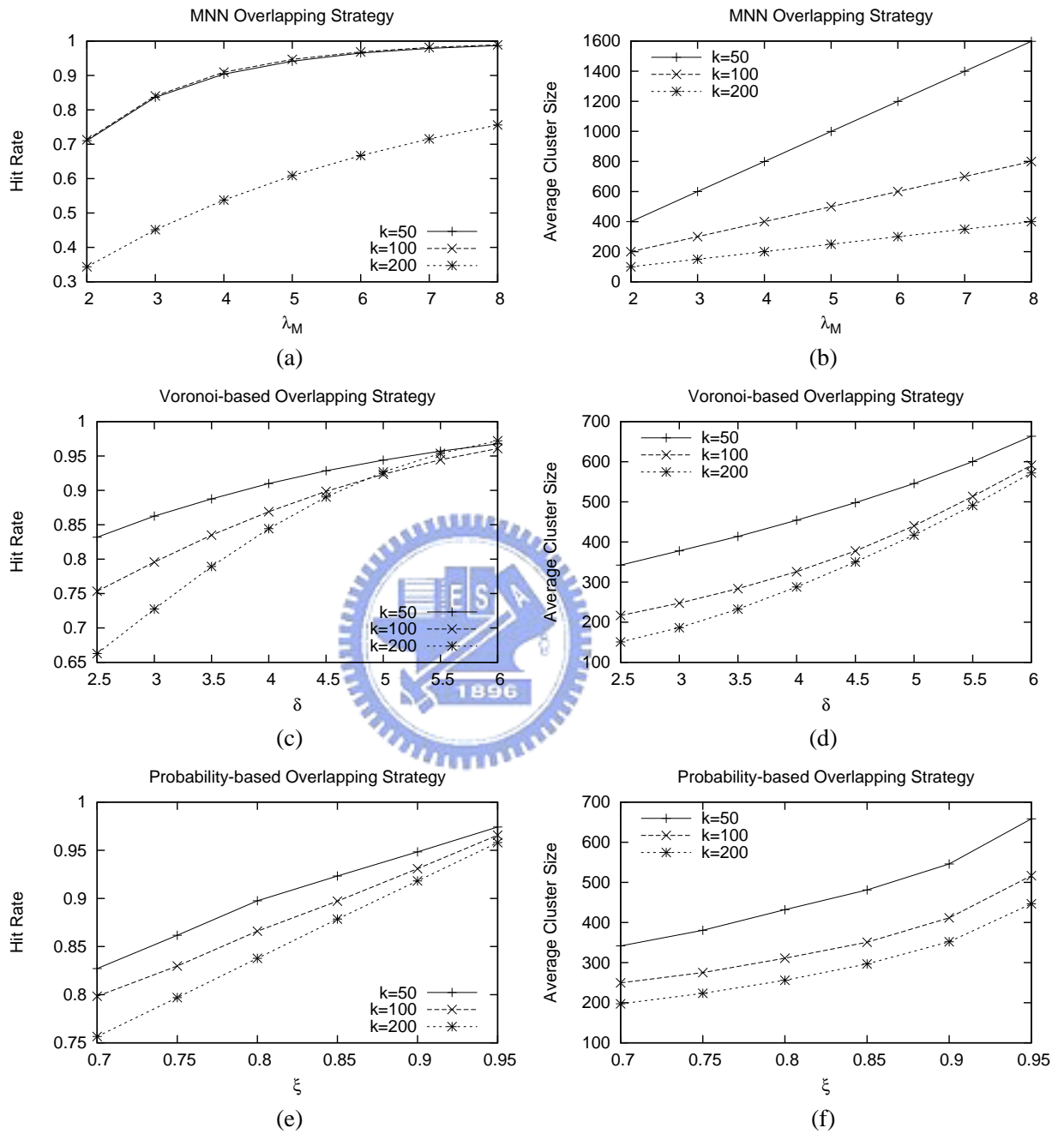


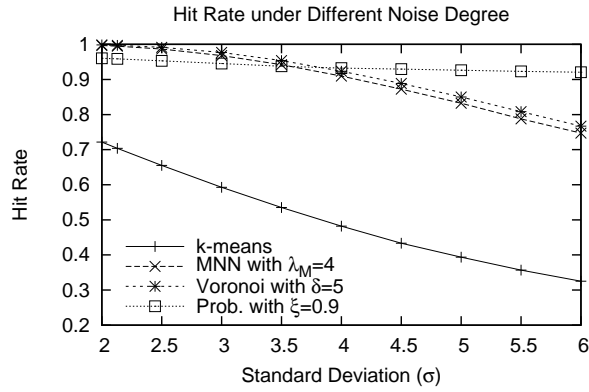
Figure 5.2: Sensitivity performance studies for three clustering strategies.

5.4 Performance Comparison of Clustering Strategies

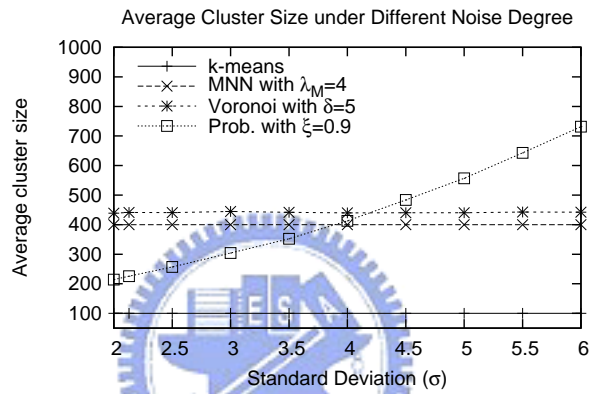
In light of sensitive performance studies in Section 5.3, we select MNN with $\lambda_M = 4$, Voronoi with $\delta = 5$, and Prob with $\xi = 0.9$ to further compare their performance with the noise degree varied. The performance study of these clustering strategies is shown in Fig. 5.3. Fig. 5.3(a) shows that in a very noisy environment (i.e., larger standard deviation), the Prob strategy outperforms the others. Also, the hit rate of MNN and Voronoi is very high when $\sigma \leq 3.5$. However, the hit rates of these two strategies decrease quickly as noise degree increases. It is worth mentioning that without the overlapping technique, k -means performs worst in terms of hit rates. However, from the result in Fig. 5.3(b), k -mean has the smallest average cluster size. If the latency caused by positioning is more important, this algorithm is a good choice because smaller average cluster size implying shorter latency. Besides, both MNN and Voronoi have reasonable average cluster size in the environment with larger noise degree. On the other hand, if accuracy of location estimation is more important, one should employ the Prob strategy. Hence, upon the requirement of applications, one should determine to use a suitable clustering technique.

5.5 Performance Study of Total Comparison Cost

The number of clusters will impact on the computation cost. Hence, we further conduct some experiments by varying the number of clusters. The experimental result is shown in Fig. 5.4, where the total comparison cost is defined as the summation of the cluster number and the average cluster size. In this experiment, each strategy should guarantee that the hit rate is larger than 0.85. From Fig. 5.4, when $k = 100$, the total costs of Voronoi and Prob are minimal. On the other hand, MNN generates more overhead than other two strategies.



(a)



(b)

Figure 5.3: Effect of the standard deviation σ on the (a) hit rate and (b) average cluster size.

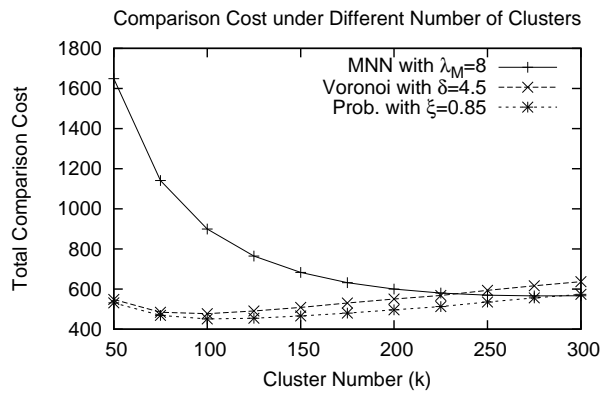


Figure 5.4: The total comparison cost under different number of clusters.

Chapter 6

Conclusion

In this paper, we presented an efficient cluster-enhanced localization framework to speed up the pattern-matching positioning algorithms in large-scale wireless networks. This framework can be plugged into any clustering and positioning algorithm. With the aid of clustering techniques, the training data can be divided into several groups based on their similarity defined in a specific feature space. Then, we selected the one which is most similar to the real-time received sample and only search the locations in it. Considering the problem of potential positioning errors caused by false cluster selection, three clustering strategies allowing overlaps are proposed. Our performance evaluation shows that the proposed overlapping strategies can greatly improve the hit rate of the clustering technique and reduce at least 90% computation cost.

Bibliography

- [1] P. Enge and P. Misra, “Special Issue on Global Positioning System,” *Proc. IEEE*, vol. 87, no. 1, pp. 3–15, 1999.
- [2] R. Want, A. Hopper, V. Falcão, and J. Gibbons, “The Active Badge Location System,” *ACM Trans. on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.
- [3] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-support System,” in *IEEE/ACM MOBICOM*. ACM Press New York, NY, USA, 2000, pp. 32–43.
- [4] P. Bahl and V. N. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System,” in *IEEE INFOCOM*, 2000, pp. 775–784.
- [5] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, “A Probabilistic Approach to WLAN User Location Estimation,” *Int’l Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.
- [6] R. Battiti, T. L. Nhat, and A. Villani, “Location-aware Computing: A Neural Network Model for Determining Location in Wireless LANs,” University of Trento, Department of Information and Communication Technology, Tech. Rep. DIT-5, 2002.
- [7] M. Brunato and R. Battiti, “Statistical Learning Theory for Location Fingerprinting in Wireless LANs,” *Computer Networks*, vol. 47, no. 6, 2005.
- [8] V. Seshadri, G. V. Záruba, and M. Huber, “A Bayesian Sampling Approach to In-door Localization of Wireless Devices using Received Signal Strength Indication,” in *IEEE PERCOM*, 2005, pp. 75–84.
- [9] J. Letchner, D. Fox, and A. LaMarca, “Large-Scale Localization from Wireless Signal Strength,” in *Proc. of the Nat’l Conf. on Artificial Intelligence (AAAI)*, 2005, pp. 15–20.
- [10] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, “Accuracy Characterization for Metropolitan-scale Wi-Fi Localization,” in *ACM MOBISYS*, vol. 5, 2005, pp. 233–245.

- [11] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-Mapping in 802.11 Location Systems," in *Proc. 7th Int'l Conf. on Ubiquitous Computing (UBICOMP)*. Springer, 2005, pp. 87–104.
- [12] 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 *IEEE/ACM MOBICOM*, 2004.
- [13] X. Chai and Q. Yang, "Reducing the Calibration Effort for Location Estimation Using Unlabeled Samples," in *IEEE PERCOM*, 2005, pp. 95–104.
- [14] J. J. Pan, J. T. Kwok, Q. Yang, and Y. Chen, "Multidimensional Vector Regression for Accurate and Low-Cost Location Estimation in Pervasive Computing," *IEEE Trans. on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1181–1193, 2006.
- [15] P. Krishnan, A. S. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu, "A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks," in *IEEE INFOCOM*, vol. 2, 2004, pp. 1001–1011.
- [16] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.
- [17] M. Youssef and A. Agrawala, "On the Optimality of WLAN Location Determination Systems," in *Comm. Networks and Dist. Syst. Modeling and Simulation Conf.*, 2004.
- [18] A. Agiwal, P. Khandpur, and H. Saran, "LOCATOR: Location Estimation System for Wireless LANs," in *ACM WMASH*, 2004, pp. 102–109.
- [19] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN Location Determination via Clustering and Probability Distributions," in *IEEE PERCOM*, 2003, pp. 143–150.
- [20] R. Xu and D. W. II, "Survey of Clustering Algorithms," *IEEE Trans. on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [21] F. Aurenhammer, "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.
- [22] T. S. Rappaport, *Wireless Communications. Principles and Practice*, 1996.