Extended Kernel-Based Location Fingerprinting in Wireless Sensor Networks

Zhansheng Duan
Center for Info Engr Science Research
Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
Email: zduan@uno.edu

Qi Zhou
Tencent Corporation
Shenzhen, Guangdong 518057, China
Email: joeyqzhou@tencent.com

Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Lab
Inst for Anthropomatics and Robotics
Karlsruhe Inst of Technology, Germany
Email: Uwe.Hanebeck@kit.edu

Abstract—**Fingerprinting localization is to estimate a mobile terminal's location using its online received signal strength (RSS) measurement and offline RSS database originated from multiple access points (APs). Kernel-based fingerprinting localization is such a competitive algorithm. However, all training data need to be considered in its offline model learning stage. This render high risks for overfitting. To alleviate this, we suggest to apply clustering to the localization region of interest first and then use kernal-based fingerprinting localization for each cluster. A byproduct of clustering is that the computational load for each cluster is also significantly reduced. To further reduce the computational load within each cluster, we also suggest to apply principal component compression to the raw RSS measurements to reduce their dimensionality. The rationale for applying principal component compression is that the distributions of the RSS measurements at all calibration points (CPs) within each cluster will be more similar after clustering. Performance evaluation using both simulated data and real data show that the extended kernel-based fingerprinting localization using clustering and principal component compression have better location estimation accuracy and less computational load.**

Keywords: Location fingerprinting, received signal strength, overfitting, clustering, principal component compression, dimensionality reduction.

I. Introduction

Localization of a mobile device has received much attention recently with the development of smart mobile devices, pervasive computing, and location based service. Global Positioning System (GPS) works well in outdoor environments. However, it may fail in urban or indoor environments. Many model-based localization algorithms have been developed utilizing the geometrical relationship between an access point (AP) and a mobile terminal, such as time difference of arrival (TDOA), RSS, and angle of arrival (AOA). However, these measurements can suffer from complex signal propagation, especially in indoor environments.

The widespread use of WLAN has made the localization algorithm using RSS measurements practical in many indoor environments. As an alternative to model-based localization [1], LF is a model-free method using RSS measurements in that it does not make any assumptions on the measurement model [2]. Many LF algorithms have been investigated in recent years. K-nearest neighbor (KNN) [3] was used in user location and tracking system in indoor environments. A probabilistic approach to estimate user location by WLAN was proposed in [4]. It has presented a probabilistic framework for LF problem. Statistical learning theory provides a profound theoretical basis for LF prolbem. Many statistical learning algorithms like support vector regression [5], [6], weighted KNN, and neural network [7] for LF problem have been investigated recently. From filtering perspective, Bayesian and Kalman filters were applied to LF in [8]. In addition to the current measurements, previous measurements are also considered for the current location estimate. In [9] strategies to generate a subset of calibration points (CPs) and AP were proposed. Also, a kernelized measure for evaluation of similarity between an RSS vector and the training RSS records was proposed.

A kernel regression based method for LF (KLF) was proposed in [10]. This method is quite competitive. However, with an increase of the number of calibration points (CPs) and APs, the computational complexity of KLF and its memory requirements will increase dramatically. The mobile terminal is usually small in size and thus has limited battery capacity. Also, location aware services usually need to access the location as fast as possible. For the KLF method, how to reduce the computational complexity without sacrificing the mobile terminal's location estimation accuracy too much is a challenging problem.

To tackle this, an idea is to reduce the dimensionality of the RSS measurements. This was achieved first by Youssef et al. in [11] by choosing a subset of the APs with the strongest signal. However, the APs with the strongest signal are not necessarily the most discriminating ones. An information gain based [12] AP selection strategy for radio map based LF was proposed in [13]. The information gain is designed only for classification purpose but not for regression purpose. Therefore only the discrete cell or grid index rather than the continuous location can be determined.

The number of APs is in fact the dimension of the RSS measurements. Some transformation methods, e.g., PCA [14], ICA, DCT, were proposed to reduce the dimensionality of RSS

Research supported in part by National 973 project of China through grant 2013CB329405, National Natural Science Foundation of China through grant 61174138, and the Fundamental Research Funds for the Central Universities of China.

data in [15], [16]. It was found that PCA LF outperforms DCT LF and ICA LF. The algorithm that applies PCA to the KLF is called PCA KLF. However, the distributions of RSS measurements at different CPs are different, especially when the CPs are far away from each other. So the implicit underlying assumption that the distributions of the RSS observations from all CPs are the same when applying the PCA is violated in [15], [16].

Cluster analysis can be used for LF to reduce the computational load. In [17], proximity graphs were employed for predicting performance of LF algorithm. This also eliminates unnecessary fingerprints to reduce computational load. In [17], cluster analysis was used and proximity graphs were applied only to each small cluster. In [18], artificial neural network model was used to group the CPs into clusters. In the estimation phase, Kohonen networks as a type of self organizing map was used to convert high-dimensional RSS into a two-dimensional discrete map. The method proposed in [19] groups CPs into several clusters according to its highest and second highest RSSs. It allows that a CP can have more than one fingerprint and can be grouped into multiple clusters. In [20], to avoid the false cluster selection in the online stage, several clustering strategies to enhance the k-means algorithm by allowing clusters to have overlapping members were proposed.

In this paper, we suggest the use of two new LF algorithms called mKLF and mPCA KLF to extend the KLF algorithm. The mKLF uses cluster analysis to group the training data into several clusters. This can help alleviate the potential overfitting problem of KLF and reduce the computational complexity in the offline stage. PCA is then also suggested to compress the original RSS measurement to a relatively lower dimension. This can further reduce the computational complexity. It should be also noted that cluster analysis has the byproduct to make the underlying assumption when applying PCA, i.e., all the involved RSS measurements should follow the same distribution, more valid. Therefore the use of PCA is more reasonable after cluster analysis. Illustrative examples show that the newly suggested mKLF and mPCA KLF outperform the KLF and PCA KLF.

The rest of this paper is organized as follows. Section II formulates the LF problem. Section III gives a brief summary of KLF. Section IV discusses the new mKLF and mPCA KLF in detail. Section V provides evaluations to mKLF and mPCA KLF using both simulated data and real data. Section VI concludes the paper.

II. Problem formulation

Suppose that we have M APs and N CPs. The augmented RSS measurement at the l-th CP is denoted as $\mathbf{r}_l$, where

$$\mathbf{r}_l = [r_l^1, r_l^2, \cdots, r_l^M]^T, \ l = 1, \ldots, N,$$

and r_l^i is the RSS measurement with respect to the i-th AP, $i = 1, \ldots, M$. The offline RSS data set is denoted as

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N].$$

The location of the l-th CP is denoted as $\mathbf{p}_{\text{CP}_l}$, where

$$\mathbf{p}_{\text{CP}_l} = [p^x_{\text{CP}_l}, p^y_{\text{CP}_l}]^T \in \mathbb{R}^2, \ l = 1, \ldots, N$$

is its coordinates in 2D Cartesian plane.

In LF, one needs to train a mapping $\psi(\cdot)\colon \mathbb{R}^M \to \mathbb{R}^2$ from the RSS measurement space to the 2D location space using the offline training data set $\{(\mathbf{r}_l, \mathbf{p}_{\text{CP}_l})\}_{l=1}^N$ first. Then for online location applications, one can simply use the obtained mapping $\psi(\cdot)$ and the real-time RSS measurement of a mobile terminal to determine its location.

III. Summary of kernel-based fingerprinting localization

It was shown in [10] that the kernel-based method is more accurate than the popular WKNN method for LF. Due to limited space, only the kernel-based ridge regression for LF will be considered in this paper.

In the kernel-based method, one maps the original feature space $\mathbf{r}_l, \ \ l = 1, \cdots, N$ into a nonlinear feature space $\phi(\mathbf{r}_l)$ by the kernel function

$$k(\mathbf{r}_l, \mathbf{r}_j) = \phi(\mathbf{r}_l)\phi(\mathbf{r}_j) \tag{1}$$

The general idea of kernel trick is that if an algorithm can be formulated only by the inner product, if we replace the inner product $\mathbf{r}_l \cdot \mathbf{r}_j$ by $\phi(\mathbf{r}_l)\phi(\mathbf{r}_j)$, then the original feature space is implicitly mapped into a higher dimensional one. Explicit representation for $\phi(\cdot)$ is not required and the inner product $\phi(\mathbf{r}_l)\phi(\mathbf{r}_j)$ can be replaced by the kernel function $k(\mathbf{r}_l, \mathbf{r}_j)$.

The cost function for LF can be written as:

$$c(\psi_d) = \frac{1}{N}\sum\nolimits_{l=1}^N (p^d_{\text{CP}_l} - \psi_d(\mathbf{r}_l))^2 + \eta \left\|\psi_d\right\|^2_{\mathcal{H}}, \tag{2}$$

where $d \in \{x, y\}$, $[\psi_x(\cdot), \psi_y(\cdot)]^T = \psi(\cdot)$, and η is a positive tuning parameter that controls the trade-off between fitness error and the complexity of the solution.

According to the reproducing property, the minimizer of the cost function (2) is of the following general form

$$\psi_d(\cdot) = \sum\nolimits_{l=1}^N \alpha_{l,d}\kappa(\mathbf{r}_l, \cdot), \tag{3}$$

where $\kappa(\mathbf{y}_l, \cdot)$ is a reproducing kernel and $\alpha_{l,d}$ is the corresponding ridge regression coefficient.

By substituting (3) into (2), the following dual optimization problem in terms of α_d can be obtained

$$\min_{\alpha_d}(\mathbf{p} - K\alpha_d)^T(\mathbf{p} - K\alpha_d) + \eta N \alpha_d^T K \alpha_d \tag{4}$$

where $\alpha_d = [\alpha_{1,d}, \cdots, \alpha_{N,d}]^T$, $\mathbf{p}_d = [p^d_{\text{CP}_1}, \cdots, p^d_{\text{CP}_N}]^T$, $\mathbf{I}_N$ is an N-dimensional identity matrix, $\mathbf{K} = [\mathbf{K}_{l,j}]_{l,j=1}^N$ is a kernel matrix with $\mathbf{K}_{l,j} = \kappa(\mathbf{y}_l, \mathbf{y}_j)$. In this paper, the Gaussian kernel

$$\kappa(\mathbf{r}_l, \mathbf{r}_j) = \exp(-\frac{||\mathbf{r}_l - \mathbf{r}_j||^2}{2\sigma_\kappa^2})$$

is considered, where σ_κ^2 determines the width of the Gaussian kernel.

The solution of (4) is of the following form

$$\alpha_d = (\mathbf{K} + \eta N \mathbf{I}_N)^{-1}\mathbf{p}_d, \quad d \in \{x, y\}, \tag{5}$$

For the location fingerprinting problem, given the coefficient vector α_d and the mapping (3), one can easily obtain the location estimate of the mobile terminal from its RSS measurement.

Remark 1: From (3), it can be seen that due to the involvement of all available RSS measurements in the database for regression, the KLF method may still be subject to high risks for overfitting although η has controlled this to certain extent.

IV. Extended kernel-based fingerprinting localization

A. Extension using clustering only

The computational complexity of the KLF and its memory requirements will increase dramatically with an increase of the number of CPs and APs. It is therefore preferred to reduce the computational complexity without sacrificing the mobile terminal's location estimation accuracy too much. From Eq. (3), we see that all training data are used during the offline stages. With the increase of the training data, i.e., the number of CPs, the mapping becomes more and more complicated, which may render high risks for overfitting. To alleviate the potential overfitting problem, a method called multiple kernel based location fingerprinting (mKLF) is suggested. It first divides all the training data into small clusters. Then for each cluster, the KLF is applied. Because only the training data within each cluster is used, it reduces the fingerprinting comparison significantly. Thus the computational complexity can be reduced. Although the number of training data in each cluster is reduced, the CPs in each cluster are more related.

The workflow of the mKLF is shown in Fig. 1. It can be seen that the workflow consists of the following two stages.

Off-line stage:

Step 1: Acquire the original RSS data set $\mathbf{R}$ from N CPs.

Step 2: Apply a cluster analysis method, e.g., k-means, to the positions of all available CPs to get C clusters of the localization region of interest. After this, the training data subset for the i-th cluster is $\mathbf{R}^i = [\mathbf{r}_{i_1}, \mathbf{r}_{i_2}, \cdots, \mathbf{r}_{i_{N_i}}]$, where N_i is the number of CP's belonging to it, and $\mathbf{r}_{i_j}$, $j = 1, \ldots, N_i$ is the j-th RSS measurement belonging to it.

Step 3: Calculate the sample mean of all RSS measurements of the i-th cluster

$$\bar{\mathbf{r}}_i = \frac{1}{N_i}\sum\nolimits_{j=1}^{N_i} \mathbf{r}_{i_j}, \quad i = 1, \ldots, C.$$

Step 4: For the i-th cluster, use the input-output data pair $(\mathbf{r}_{i_j}, \mathbf{p}_{\mathrm{CP}_{i_j}})$, $j = 1, \ldots, N_i$, and an KLF algorithm to train a mapping $\psi^i(\cdot)$.

On-line stage:

Step 1: Acquire the RSS measurement $\mathbf{r}_{mt}$ from all N APs.

Step 2: Find the cluster to which the mobile terminal belongs according to

$$i^* = \arg \min_{i \in \{1,\ldots,C\}} d(\mathbf{r}_{mt}, \bar{\mathbf{r}}_i),$$

where $d(\cdot, \cdot)$ is the Euclidean distance between two RSS measurements.

Step 4: Apply the decided mapping $\psi^{i^*}(\cdot)$ to obtain the estimated location $\hat{\mathbf{p}}_{mt}$ of the mobile terminal as

$$\hat{\mathbf{p}}_{mt} = \psi^{i^*}(\mathbf{r}_{mt}).$$

Remark 2: For simplicity, the cluster analysis is over the positions but not the RSS measurements of all available CPs. We agree that the use of RSS measurements for cluster analysis is more reasonable since they also account for the impact of the environments over the measurement. However, it is not used here for cluster analysis because of two reasons. First, closely located CPs will have close RSS measurements in general. Second, computation-wise the use of RSS measurement is not preferred because the location is just a two-dimensional vector but the RSS measurement is M-dimensional.

B. Extension using both clustering and principal component compression

If we want to further reduce the computation of mKLF without sacrificing the accuracy too much, an idea to achieve this trade-off is to find a new dimensionality-reduced vector $\mathbf{y}_l \in \mathbb{R}^L$, where $L < M$, to represent the original RSS measurement $\mathbf{r}_l$. Then $\mathbf{y}_l$ can be used as the new input data to the mKLF algorithms.

In [16], PCA was used to reduce the dimension of the original RSS measurement. First by subtracting the sample mean from each column, the original input data set $\mathbf{R}$ is changed to a new zero-mean input data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, where $\mathbf{x} = \mathbf{r} - \bar{\mathbf{r}}$ and $\bar{\mathbf{r}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{r}_j$. Then one can use PCA to linearly compress the data set $\mathbf{X}$ to a dimensionality-reduced one $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}, \tag{6}$$

where

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N].$$

PCA can be defined [21] as the linear projection onto the principal subspace that minimizes the average squared projection error

$$J = \frac{1}{N}\sum\nolimits_{i=1}^{N} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2,$$

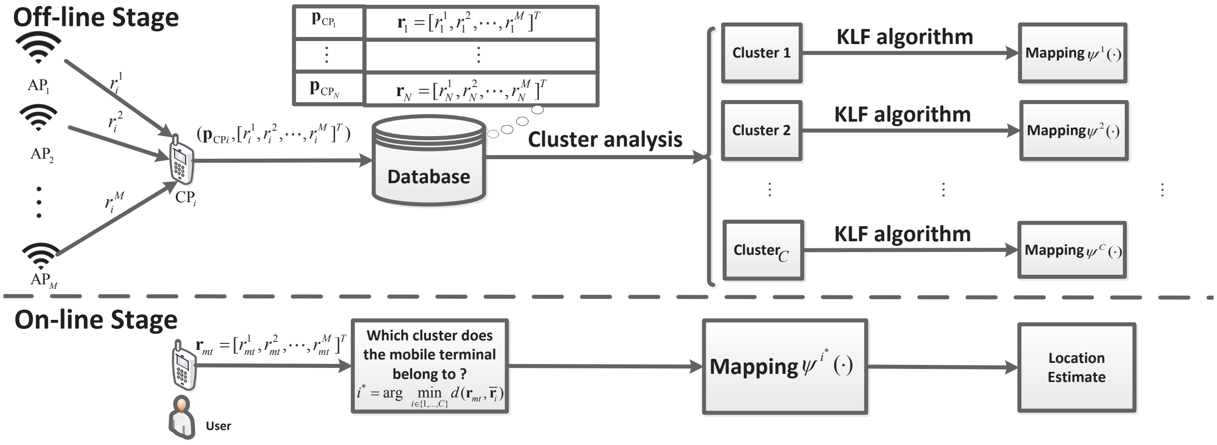
where

$$\hat{\mathbf{x}}_i = \sum\nolimits_{j=1}^{L} \alpha_{j,i}\mathbf{u}_j + \sum\nolimits_{k=L+1}^{M} \beta_k \mathbf{u}_k,$$

and L is a given desired dimension for the principal subspace, $\{\mathbf{u}_j\}_{j=1}^{M}$ is a complete orthonormal set of M-dimensional basis vectors, and $\alpha_{j,i}$ depends on $\mathbf{x}_i$ but β_k does not.

It was found that when $[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_L]$ are the eigenvectors corresponding to the first L largest eigenvalues of the sample covariance matrix $\mathbf{P}$ of $\{\mathbf{x}_i\}_{i=1}^{N}$, i.e.,

$$\mathbf{P} = \frac{1}{N}\sum\nolimits_{i=1}^{N}(\mathbf{r}_i - \bar{\mathbf{r}})(\mathbf{r}_i - \bar{\mathbf{r}})^T = \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T \tag{7}$$



Off-line Stage
AP_1
AP_2
AP_M
r_i^1
r_i^2
r_i^M
CP_i
$(\mathbf{p}_{\mathrm{CP}i},[r_i^1,r_i^2,\cdots,r_i^M]^T)$
$\mathbf{p}_{\mathrm{CP}_1}$
$\mathbf{r}_1=[r_1^1,r_1^2,\cdots,r_1^M]^T$
$\mathbf{p}_{\mathrm{CP}_N}$
$\mathbf{r}_N=[r_N^1,r_N^2,\cdots,r_N^M]^T$
Database
Cluster analysis
Cluster 1
Cluster 2
Cluster C
KLF algorithm
KLF algorithm
KLF algorithm
Mapping $\psi^1(\cdot)$
Mapping $\psi^2(\cdot)$
Mapping $\psi^C(\cdot)$
On-line Stage
$\mathbf{r}_{mt}=[r_{mt}^1,r_{mt}^2,\cdots,r_{mt}^M]^T$
User
Which cluster does the mobile terminal belong to ?
$i^*=\arg\min_{i\in\{1,\ldots,C\}} d(\mathbf{r}_{mt},\overline{\mathbf{r}}_i)$
Mapping $\psi^{i^*}(\cdot)$
Location Estimate

Figure 1: Workflow of mKLF

and $[\mathbf{u}_{L+1},\mathbf{u}_{L+2},\cdots,\mathbf{u}_M]$ are the eigenvectors corresponding to the $M-L$ smallest eigenvalues of $\mathbf{P}$, the projection cost achieves its minimum. It was also found that

$$\alpha_{j,i}=\mathbf{x}_i^T\mathbf{u}_j,\ \beta_k=\bar{\mathbf{x}}^T\mathbf{u}_k.$$

For the problem we are considering, $\beta_k=0$ because $\bar{\mathbf{x}}=0$. Thus an L-dimensional compression to the original data is simply

$$\mathbf{y}_i=\mathbf{A}\mathbf{x}_i,$$

where

$$\mathbf{A}=[\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_L]^T.$$

Remark 3: An implicit assumption of the PCA LF is that the distributions of the RSS measurements from different CPs are the same. This can be clearly seen from (7) for sample covariance $\mathbf{P}$ which requires that all samples $\{\mathbf{r}_i\}_{i=1}^N$ should be generated from the same distribution. However, this underlying assumption is hardly true in practice especially for the RSS measurements from CPs far away form each other. For example, the means of the RSS measurements from apart CPs are significantly different when the Okumura-Hata model [10] is used.

As for the mKLF method, compared to the whole training data, the training data within each cluster is more similar, which has alleviated the violation of the underlying assumption of PCA. So a method called multiple PCA kernel based location fingerprinting is suggested. It first apply cluster analysis to the positions of all CPs so that the localization region of interest is partitioned into certain number of clusters. Then a PCA based transformation is applied to each cluster. Then the KLF method is applied to each cluster. Compared with the use of all training data, cluster analysis makes the number of training data in each cluster significantly reduced. So it helps alleviate the overfitting problem. Also, a byproduct of its is that the distributions of the RSS measurements within the same cluster will not be that different. Thus the underlying assumption of PCA can be satisfied to certain extent. This is similar to the hybrid grid scheme [22] for estimation problem with model/parameter uncertainty.

The workflow of the mPCA KLF is shown in Fig. 2. It can be seen that the workflow consists of the following two stages.

Off-line stage:

Step 1: Acquire the original RSS data set $\mathbf{R}$ from N CPs.

Step 2: Apply a cluster analysis method, e.g., k-means, to the positions of all available CPs to get C clusters of the localization region of interest. After this, the training data subset for the i-th cluster is $\mathbf{R}^i=[\mathbf{r}_{i_1},\mathbf{r}_{i_2},\cdots,\mathbf{r}_{i_{N_i}}]$, where N_i is the number of CP's belonging to it, and $\mathbf{r}_{i_j},\ j=1,\ldots,N_i$ is the j-th RSS measurement belonging to it.

Step 3: Calculate the sample mean of all RSS measurements of the i-th cluster

$$\bar{\mathbf{r}}_i=\frac{1}{N_i}\sum\nolimits_{j=1}^{N_i}\mathbf{r}_{i_j},\quad i=1,\ldots,C.$$

Step 4: For the i-th cluster, $i=1,\ldots,C$, subtract the sample mean from all its measurements to obtain the corresponding zero-mean data set $\mathbf{X}^i=[\mathbf{x}_{i_1},\mathbf{x}_{i_2},\cdots,\mathbf{x}_{i_{N_i}}]$, where

$$\mathbf{x}_{i_j}=\mathbf{r}_{i_j}-\bar{\mathbf{r}}_i.$$

Step 5: Apply PCA to each data subset $\mathbf{X}^i$, $i=1,\ldots,C$ to obtain a transformation matrix $\mathbf{A}_i$ so that each piece of original data $\mathbf{x}_{i_j}$ is reduced to a lower dimensional one $\mathbf{y}_{i_j}$, $j=1,\ldots,N_i$, through

$$\mathbf{y}_{i_j}=\mathbf{A}_i\mathbf{x}_{i_j},\ i=1,\ldots,C,\ j=1,\ldots,N_i.$$

Step 6: For the i-th cluster, use the input-output data pair $(\mathbf{y}_{i_j},\mathbf{p}_{\mathrm{CP}_{i_j}}),\ j=1,\ldots,N_i$, and an KLF algorithm to train a mapping $\psi^i(\cdot)$.

On-line stage:

Step 1: Acquire the RSS measurement $\mathbf{r}_{mt}$ from all N APs.

Step 2: Find the cluster to which the mobile terminal belongs according to

$$i^*=\arg\min_{i\in\{1,\ldots,C\}} d(\mathbf{r}_{mt},\bar{\mathbf{r}}_i),$$

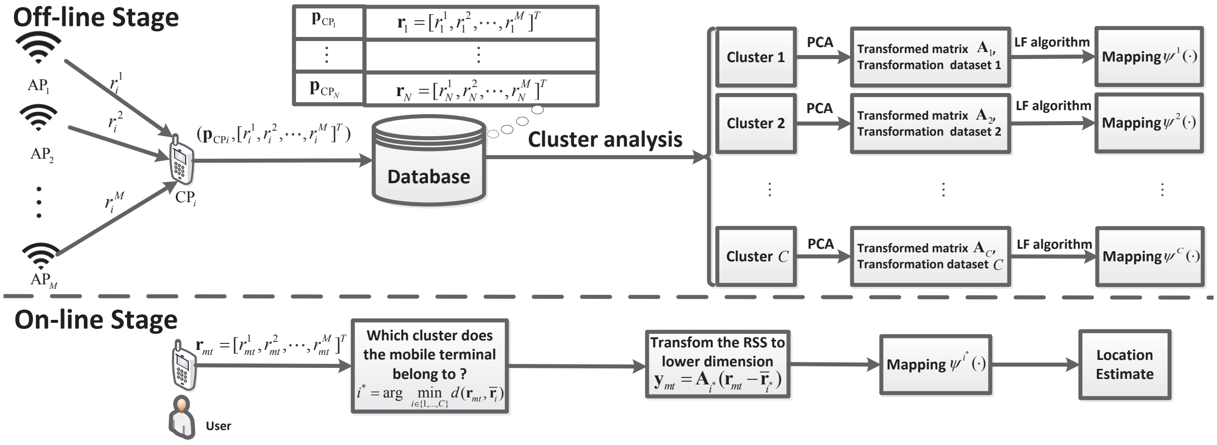

Off-line Stage
AP$_1$
AP$_2$
AP$_M$
r_i^1
r_i^2
r_i^M
CP$_i$
$(\mathbf{p}_{\mathrm{CP}i},[r_i^1,r_i^2,\cdots,r_i^M]^T)$
$\mathbf{p}_{\mathrm{CP}_1}$
$\mathbf{r}_1=[r_1^1,r_1^2,\cdots,r_1^M]^T$
$\mathbf{p}_{\mathrm{CP}_N}$
$\mathbf{r}_N=[r_N^1,r_N^2,\cdots,r_N^M]^T$
Database
Cluster analysis
Cluster 1
Cluster 2
Cluster C
PCA
Transformed matrix $\mathbf{A}_1$, Transformation dataset 1
Transformed matrix $\mathbf{A}_2$, Transformation dataset 2
Transformed matrix $\mathbf{A}_C$, Transformation dataset C
LF algorithm
Mapping $\psi^1(\cdot)$
Mapping $\psi^2(\cdot)$
Mapping $\psi^C(\cdot)$
On-line Stage
$\mathbf{r}_{mt}=[r_{mt}^1,r_{mt}^2,\cdots,r_{mt}^M]^T$
User
Which cluster does the mobile terminal belong to ?
$i^*=\arg\min_{i\in\{1,\ldots,C\}} d(\mathbf{r}_{mt},\overline{\mathbf{r}}_i)$
Transfom the RSS to lower dimension
$\mathbf{y}_{mt}=\mathbf{A}_{i^*}(\mathbf{r}_{mt}-\overline{\mathbf{r}}_{i^*})$
Mapping $\psi^{i^*}(\cdot)$
Location Estimate

Figure 2: Workflow of mPCA KLF

where $d(\cdot,\cdot)$ is the Euclidean distance between two RSS measurements.

Step 3: Use the the corresponding transformation matrix $\mathbf{A}_{i^*}$ to obtain the lower-dimensional input

$$\mathbf{y}_{mt}=\mathbf{A}_{i^*}(\mathbf{r}_{mt}-\overline{\mathbf{r}}_{i^*}).$$

Step 4: Apply the decided mapping $\psi^{i^*}(\cdot)$ to obtain the estimated location $\hat{\mathbf{p}}_{mt}$ of the mobile terminal as

$$\hat{\mathbf{p}}_{mt}=\psi^{i^*}(\mathbf{y}_{mt}).$$

V. Illustrative examples

For performance evaluation purpose, both simulated data and real data have been used. To demonstrate performance improvement, mKLF and mPCA KLF are compared with KLF and PCA KLF.

First, the comparison is conducted using simulated data. Consider a 100 m × 100 m rectangle region with 16 APs and 400 CPs uniformly distributed over it as in Fig. 3.

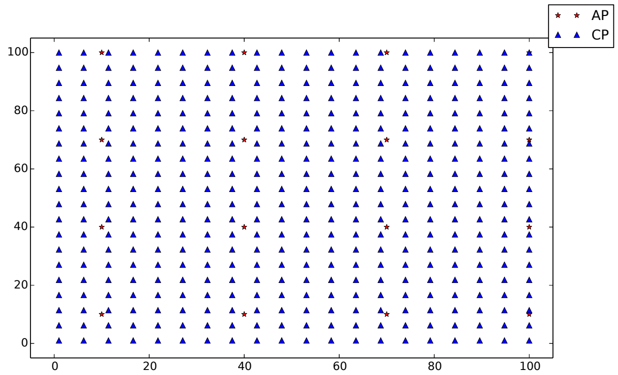


Figure 3: The deployment of APs and CPs for simulated data

The RSS measurement is generated from the Okumura-Hata model [10]

$$r_l^i=r_0-10n_P\log_{10}||\mathbf{p}_{\mathrm{AP}_i}-\mathbf{p}_{\mathrm{CP}_l}||\ +\varepsilon_l^i, \tag{8}$$

where r_l^i is the RSS measurement at the l-th CP originated from the i-th AP, r_0 is the initial power (set to $150dBm$), $||\mathbf{p}_{\mathrm{AP}_i}-\mathbf{p}_{\mathrm{CP}_l}||$ is the Euclidian distance between the i-th AP and the l-th CP, n_P is the path-loss exponent (set to 4), and ε_l^i is the accompanying Gaussian measurement noise with zero mean and variance σ_ε^2.

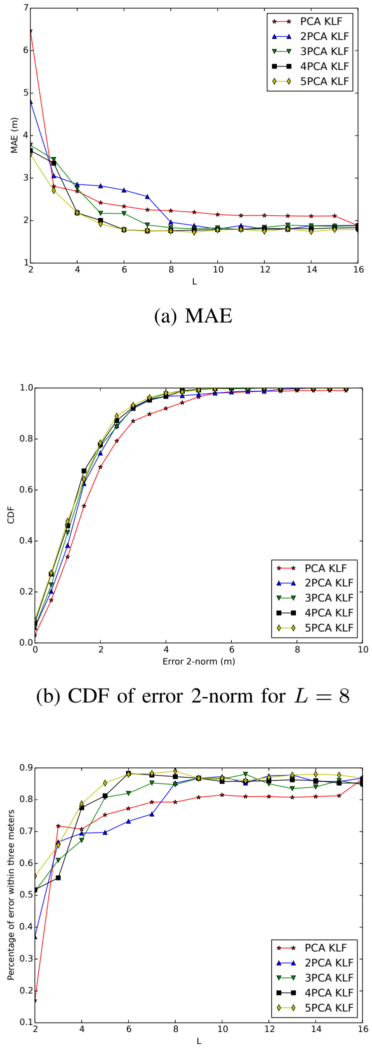
Leave-one-out cross validation is a valid way for parameter selection and performance evaluation. In leave-one-out cross validation, one RSS measurement is chosen as the testing set and the remaining ones are chosen as the training set. Repeat this procedure until each RSS measurement has been chosen as the testing set. Using leave-one-out cross validation, it is obtained that the best η and σ_κ are $\eta=2^{-20}$ and $\sigma_\kappa=2^7$ for the simulated data. In mPCA KLF, we choose the k-means for cluster analysis and set $C=2,3,4,5$. KLF and PCA KLF are chosen to be compared with mPCA KLF and mKLF.

The mean absolute error (MAE) [23] is a frequently used measure to evaluate the estimation accuracy. For LF problem, it is defined as:

$$\mathrm{MAE}(\hat{\mathbf{p}}_{mt})=\frac{1}{N_t}\sum_{i=1}^{N_t}|\hat{\mathbf{p}}_{mt}^i-\mathbf{p}_{mt}| \tag{9}$$

where $\hat{\mathbf{p}}_{mt}^i$ is the i-th location estimate of leave-one-out cross validation and N_t is the total number of estimation times.

For the first simulated scenario, the variance of the additive Gaussian noise σ_ε^2 is set to 1. The MAEs of mPCA KLFs ($C=2,3,4,5$) and PCA KLF obtained using leave-one-out cross validation are shown in Fig. 4a. When $L=16$, the dimensionality reduced technique PCA is not necessary. So when $L=16$, the MAEs of PCA KLF and mPCA KLF are reduced to those of KLF and mKLF as shown in Fig. 4a. It can be seen that the mPCA KLFs are all more accurate than PCA KLF when the dimension of the transformed data is larger than or equal to eight. Also, mPCA KLF is more accurate than KLF which has used all training data. This means that mPCA KLF has not only reduced the computational load of KLF but also increased its accuracy. When $8\le L\le 15$ mPCA KLF has similar accuracy to that of mKLF. But mPCA KLF has lower computational load than mKLF which will be illustrated next. From another perspective, the percentage of the points whose estimation error is within 3m is shown in Fig.4c. It can be seen that all mPCA KLFs have more accurate points than PCA KLF when the $L \geq 8$. It can be further seen from Fig. 4a that $L = 8$ is the best choice for the tradeoff between computational load and location estimation accuracy. The cumulative distribution function (CDF) of location estimation error 2-norm for $L = 8$ is shown in Fig. 4b. It can be clearly seen that the CDF's of all mPCA KLFs are above that of PCA KLF. Also, the more clusters the region of interest is partitioned, the better the location estimation accuracy.



(a) MAE

(b) CDF of error 2-norm for $L = 8$

(c) Percentage of points with error within 3m

Figure 4: Simulated data with $\sigma_\varepsilon^2 = 1$

In practical applications, the computational load of the online stage plays an important role in the LF method performance. So the computational load of the online stage of each method is evaluated. Because multiplication takes more time than addition, only multiplications will be considered for computational load evaluation. For KLF, it needs to consider all N training data. Each data is M-dimensional. In total KLF needs $N \cdot M$ multiplications. For PCA KLF, the dimension of the data is reduced to L, but it needs $L \cdot M$ extra multiplications for the data transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$. So in total PCA KLF needs $N \cdot L + L \cdot M$ multiplications. For mKLF on average it takes $\frac{N}{C} \cdot M + C \cdot M$ multiplications, where $C \cdot M$ multiplications are from the decision process $i^* = \arg \min_{i \in \{1,\ldots,C\}} d(\mathbf{r}_{mt}, \bar{\mathbf{r}}_i)$. For mPCA KLF, on average it needs $\frac{N}{C} \cdot L + C \cdot L \cdot M + C \cdot M$ multiplications. The number of multiplications every method needs is summarized in Table I.

For this scenario, we also compared the relative python code running time of the online stage for each method. Dividing the absolute running time of the online stage for each method by that of the KLF, we have the relative running time of the online stage for each method as in Table II.

The second simulated scenario, the variance of the additive Gaussian noise σ_ε^2 is set to 4. The MAEs of mPCA KLFs ($C = 2, 3, 4, 5$) and PCA KLF obtained using leave-one-out cross validation are shown in Fig. 4a. When $L = 16$ the MAEs of PCA KLF and mPCA KLF are reduced to those of KLF and mKLF. It can be seen that the mPCA KLFs are all more accurate than PCA LF when the dimension of the transformed data is larger than or equal to eight. Also, mPCA KLF is more accurate than KLF which has used all training data. This means that mPCA KLF has not only reduced the computational load of KLF but also increased its accuracy. From another prospective, the percentage of the points whose estimation error is within 3m is shown in Fig.5c. It can be seen that all mPCA KLFs have more accurate points than PCA KLF and KLF when the $L \geq 8$. It can be further seen from Fig. 5a that $L = 8$ is the best choice for the tradeoff between computational load and location estimation accuracy. The CDF of location estimation error 2-norm for $L = 8$ is shown in Fig. 5b. It can be clearly seen that the CDF's of all mPCA KLFs are above that of PCA KLF. Also, the more clusters the region of interest is partitioned, the better the location estimation accuracy.

Third, the comparison is also conducted using the collected real data, which was also used in [5]. The data set is available at *http://ardent.unitn.it/software/data*. Is is collected with 257 CPs and 6 APs. The MAEs of mPCA KLFs ($C = 2, 3, 4, 5$) and PCA KLF obtained using leave-one-out cross validation for this scenario are shown in Fig. 6a. When $L = 6$ the MAEs of PCA KLF and mPCA KLF are reduced to those of the KLF and mKLF. It can be seen that the mPCA KLFs are all more accurate than PCA LF for all allowable dimensions of the transformed data. The mPCA KLF has similar accuracy to that of mKLF but with lower computational load as can be seen from Table III. From another perspective, mKLF and mPCA KLF have more accurate location estimate than KLF and PCA KLF as shown in Fig. 6c. The CDF of location estimation error 2-norm for $L = 4$ is shown in Fig. 6b, a similar phenomenon as in Fig. 4b can be also observed.

Table I: The number of multiplications for each method

KLF	PCA KLF	mPCA KLF	mKLF
$N \cdot M$	$N \cdot L + L \cdot M$	$\frac{N}{C} \cdot L + C \cdot L \cdot M + C \cdot M$	$\frac{N}{C} \cdot M + C \cdot M$

Table II: Computational load comparison using simulated data

Relative running time	PCA KLF	2PCA KLF	3PCA KLF	4PCA KLF	5PCA KLF
L=8	0.8001516	0.41368265	0.29658398	0.22408827	0.19155497
L=10	0.91468995	0.4649678	0.32235735	0.25095722	0.21049691
L=12	0.97756062	0.50662387	0.34709056	0.265419	0.22551723
L=14	1.06114586	0.5429253	0.38862215	0.29810043	0.24919617
L=16	1	0.529407479094	0.43317977678	0.34684983607	0.268571803774

Table III: Computational load comparison using real data

Relative running time	PCA KLF	2PCA KLF	3PCA KLF	4PCA KLF	5PCA KLF
L=3	0.84120485	0.43932526	0.31124994	0.24732615	0.20602325
L=4	0.89765605	0.46385849	0.31520439	0.26002322	0.21938527
L=5	0.95394358	0.50514494	0.33544964	0.27740712	0.23192127
L=6	1	0.487606024385	0.340882082303	0.262983278327	0.230893590005

The choice of the dimension L of the transformed data and the number C of the clusters is very important. L determines the tradeoff between the computational load and location estimation accuracy. The smaller L is, the less the computational load and the poorer the location estimation accuracy, and vice versa. A larger C will alleviate the overfitting problem. Also it makes the RSS measurements of the CPs within the same cluster have similar distributions, which in consequence makes the underlying assumption of PCA more valid. However, if C is too large, the sample size of training data for one cluster will be too small to obtain an accurate fingerprinting mapping. Cross validation can be used to determine the values of L and C. The above illustrative examples show that for appropriate choice of L and C, mPCA KLF outperforms KLF and PCA KLF in both accuracy and computational efficiency. Also, mPCA KLF outperforms mKLF in computational efficiency. Since the cluster analysis is applied offline, it does not increase the online computation.

VI. Conclusions

In this paper, two extensions to the KLF, called mKLF and mPCA KLF, are suggested. In the off-line stage of mKLF, we first apply cluster analysis to positions of the CPs so that the localization region of interest is partitioned into certain number of clusters. As a result, the distributions of the RSS measurements from the CPs within the same cluster will be more similar. This in consequence can help alleviate the potential overfitting risk. Then an KLF algorithm is applied to each cluster to obtain a mapping rule. In the on-line stage of mKLF, the cluster to which the RSS measurement belongs to is decided first. Then the decided mapping rule is applied to locate the mobile terminal. The mPCA KLF using PCA dimensionality reduction for each cluster can further reduce the computational load. Cluster analysis makes the underlying assumption of PCA, i.e., all samples used to calculate the sample covariance should be the same, more valid. Thus the use of PCA for each cluster in mPCA KLF is more reasonable than the use of PCA for all training data in PCA KLF. Illustrative examples show that mPCA KLF significantly outperforms KLF and PCA KLF in both location accuracy and computational efficiency. Also, mPCA KLF outperforms mKLF in computational efficiency.

References

[1] M. R. Gholami, R. M. Vaghefi, and E. G. Strom, "RSS-based sensor localization in the presence of unknown channel parameters," *IEEE Transactions on Signal Processing*, vol. 61, no. 15, pp. 3752–3759, August 2013.

[2] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, November 2007.

[3] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, Tel Aviv, Israel, March 2000, pp. 775–784.

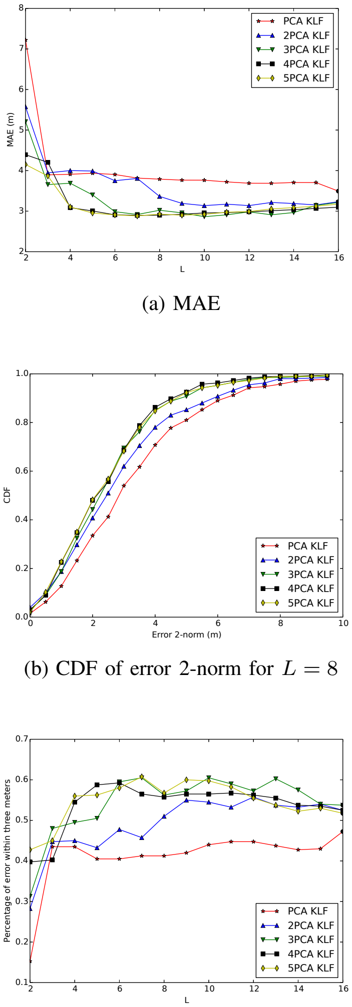
[4] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, July 2002.

[5] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825–845, April 2005.

[6] Z. L. Wu, C. H. Li, J.-Y. Ng, and K. Leung, "Location estimation via support vector regression," *IEEE Transactions on Mobile Computing*, vol. 6, no. 3, pp. 311–321, 2007.

[7] S.-H. Fang and T.-N. Lin, "Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1973 – 1978, November 2008.

[8] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piché, "A comparative survey of WLAN location fingerprinting methods," in *Proceedings of the 6th Workshop on Positioning, Navigation and Communication*, Hannover, Germany, March 2009, pp. 243–251.

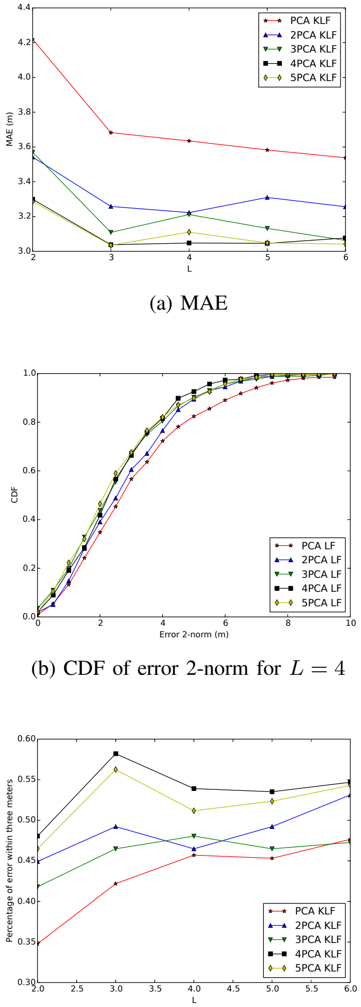


(a) MAE

(b) CDF of error 2-norm for $L = 8$

(c) Percentage of points with error within 3m

Figure 5: Simulated data with $\sigma_\varepsilon^2 = 4$



(a) MAE

(b) CDF of error 2-norm for $L = 4$

(c) Percentage of points with error within 3m

Figure 6: Real data

[9] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 689–705, June 2007.

[10] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, "Kernel-based machine learning using radio-fingerprints for localization in wsns," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 1324–1335, April 2015.

[11] M. A. Youssef, A. Agrawala, and A. Udaya Shankar, "WLAN location determination via clustering and probability distributions," in *Proceedings of the 1st IEEE International Conference on the Pervasive Computing and Communications*, Fort Worth, TX, March 2003, pp. 143–150.

[12] T. M. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[13] Y. Q. Chen, Q. Yang, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 7, pp. 877–888, July 2006.

[14] I. Jolliffe, *Principal component analysis*. Wiley, 2005.

[15] S. H. Fang, T. N. Lin, and P. Lin, "Location fingerprinting in a decorrelated space," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 685–691, May 2008.

[16] S. H. Fang and T. N. Lin, "Principal component localization in indoor WLAN environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 100–110, January 2012.

[17] N. Swangmuang and P. Krishnamurthy, "Location fingerprint analyses toward efficient indoor positioning," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, Hong Kong, March 2008, pp. 100 – 109.

[18] L. Mengual, O. Marban, and S. Eibe, "Clustering-based location in wireless networks," *Expert Systems with Applications*, vol. 9, no. 9, pp. 6165–6175, 2010.

[19] Y. Mo, Z. Cao, and B. Wang, "Occurrence-based fingerprint clustering for fast pattern-matching location determination," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2012 – 2015, DECEMBER 2012.

[20] S.-P. Kuo, B.-J. Wu, W.-C. Peng, and Y.-C. Tseng, "Cluster-enhanced techniques for pattern-matching localization systems," in *Proceedings of 2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, Pisa, October 2007, pp. 1–9.

[21] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[22] L. F. Xu, X. R. Li, and Z. S. Duan, "Hybrid grid multiple-model estimation with application to maneuvering target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 1, pp. 122–136, February 2016.

[23] R. J. Hyndman and A. B.Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.