Adaptive Localization in a Dynamic WiFi Environment Through Multi-view Learning *

Sinno Jialin Pan, James T. Kwok, Qiang Yang, and **Jeffrey Junfeng Pan**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong
{sinnopan,jamesk,qyang,panjf}@cse.ust.hk

Abstract

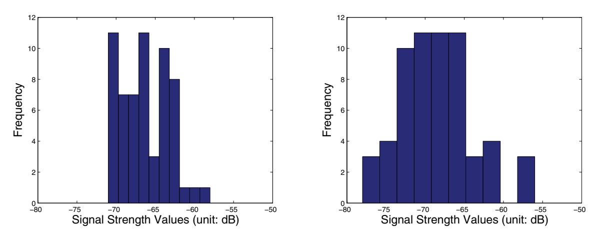
Accurately locating users in a wireless environment is an important task for many pervasive computing and AI applications, such as activity recognition. In a WiFi environment, a mobile device can be localized using signals received from various transmitters, such as access points (APs). Most localization approaches build a map between the signal space and the physical location space in a offline phase, and then using the received-signal-strength (RSS) map to estimate the location in an online phase. However, the map can be outdated when the signal-strength values change with time due to environmental dynamics. It is infeasible or expensive to repeat data calibration for reconstructing the RSS map. In such a case, it is important to adapt the model learnt in one time period to another time period without too much recalibration. In this paper, we present a location-estimation approach based on Manifold co-Regularization, which is a machine learning technique for building a mapping function between data. We describe LeManCoR, a system for adapting the mapping function between the signal space and physical location space over different time periods based on Manifold Co-Regularization. We show that LeManCoR can effectively transfer the knowledge between two time periods without requiring too much new calibration effort.We illustrate LeManCoR's effectiveness in a real 802.11 WiFi environment.

Introduction

Localizing users or mobile nodes in wireless networks using received-signal-strength (RSS) values has attracted much attention in several research communities, especially in activity recognition in AI. In recent years, different statistical machine learning methods have been applied to the localization problem (Nguyen, Jordan, & Sinopoli 2005; Ferris, Haehnel, & Fox 2006; Pan *et al.* 2006; Ferris, Fox, & Lawrence 2007).

However, many previous localization methods assume that RSS maps between the signal space and physical location space is static, thus a RSS map learnt in one time period can be applied for location estimate in latter time periods directly without adaptation. In a complex indoor WiFi environment, however, the environment is dynamic in nature, caused by unpredictable movements of people, radio interference and signal propagation. Thus, the distribution of RSS values in training and application periods may be significantly different. For example, Figure 1 shows RSS distributions at the same location in two time periods collected in our indoor environment. As a result, location estimation based on a static radio map may be grossly inaccurate.

However, collecting RSS values together with their locations is a very expensive process. Thus, it would be important for us to transfer as much knowledge from an early time period to latter time periods. If we can do this effectively, we can reduce the need to obtain new labeled data. In this paper we present LeManCoR, a localization approach that adapts a previously learned RSS mapping function for latter time periods while requiring only a small amount of new calibration data. Our approach extends the Manifold Co-Regularization of (Sindhwani, Niyogi, & Belkin 2005), a recently developed technique for semi-supervised learning with multiple views. We consider the RSS values received in different time periods at the same location as multiple views of this location. The intuition behind our approach is that the signal data in multiple views are multi-dimensional. Although their distributions are different, these data should have a common underlying low-dimensional manifold structure, which can be interpreted as the physical location space. This geometric property is the intrinsic knowledge of the localization problem in a WiFi environment.



(a) Time Period 1 (b) Time Period 2

Figure 1: Variations of signal-strength histograms in two time periods at the same location from one access point

In this paper, we extend the Manifold Co-Regularization framework to location estimation in a dynamic WiFi environment. We empirically demonstrate that this framework is effective in reducing the calibration effort when adapting the learnt mapping function between time periods. This paper contributes to machine learning by extending the Manifold Co-Regularization framework to a more general case, in which the number of data in different views can be different and there are only a subset of the data having pairwise correspondence. We also contribute to location estimation in pervasive computing, by developing a dynamic localization approach for adapting RSS mapping functions effectively.

*Sinno J. Pan, Qiang Yang and Jeffrey J. Pan are supported by NEC China Lab (NECLC05/06.EG01).

Related Work

Existing approaches to RSS based localization fall into two main categories. The first approach uses radio propagation models, which rely on the knowledge of access point locations (Bahl, Balachandran, & Padmanabhan 2000; LaMarca *et al.* 2005). In recent years, statistical machine learning methods have been applied to the localization problem. (Ferris, Haehnel, & Fox 2006) apply Gaussian process models and (Nguyen, Jordan, & Sinopoli 2005) apply Kernel methods for location estimation, respectively. In order to reduce the calibration effort, (Ferris, Fox, & Lawrence 2007) apply Gaussian-Process-Latent-Variable models (GP-LVMs) to construct RSS mapping function under an unsupervised framework. (Pan *et al.* 2006) show how to apply manifold regularization to mobile-node tracking in sensor networks for reducing calibration effort.

However, there are few previous works that consider dynamic environments with some exceptions. The LEASE system (Krishnan *et al.* 2004) utilizes different hardware systems to solve this problem. LEASE employs a number of stationary emitters and sniffers to obtain up-to-date RSS values for updating the maps. The localization accuracy can only be guaranteed when these additional hardware systems are deployed in high density. To reduce the need of the additional equipments, (Yin, Yang, & Ni 2005) apply a model tree based method, called LEMT, to adapt RSS maps by only using a few *reference points*, which are additional sensors for recording RSS values over time. LEMT needs to build a model tree at each location to capture the global relationship between the RSS values received at virous locations and those received at reference points. Another related work is (Haeberlen *et al.* 2004), which adapted a static RSS map by calibrating new RSS samples at a few known locations and fitting a linear function between these values and the corresponding values from the RSS map.

Problem Statement

Consider a two-dimensional localization problem in a dynamic indoor environment.[1] A location can be represented by $\boldsymbol{\ell} = (x, y)$. Assume that there are m access points (APs) in the indoor environment, which periodically send out wireless signals to others. The locations of APs are not necessarily known. A mobile node can measure the RSS sent by the m APs. Thus, each signal vector can be represented by $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_m)^T \in R^m$. The signal data of known locations are called labeled data, while those of unknown locations are called unlabeled data. We also assume there are l *reference points* placed at various locations for getting the real-time RSS values over different time periods. There are several time periods in each day $\{t_i\}_{i=1}^N$, where t_1 is the offline time period , in which we can collect more labeled and unlabeled data, while the others are online time periods, in which we can only collect a few labeled and unlabeled data. For each online time period t_j, $j \geq 2$, the inputs are (1) l_1 labeled data $\{(\mathbf{s}_i^{(1)}, \boldsymbol{\ell}_i^{(1)})\}_{i=1}^{l_1}$ and u_1 unlabeled data $\{\mathbf{s}_i^{(1)}\}_{i=l_1+1}^{l_1+u_1}$ collected in time period t_1, (2) a mapping function, which maps the signal vectors to locations, $f^{(1)}$ learnt in time period t_1 and (3) l labeled data $\{(\mathbf{s}_i^{(j)}, \boldsymbol{\ell}_i)\}_{i=1}^{l}$ and u_j unlabeled data $\{\mathbf{s}_i^{(j)}\}_{i=l_1+1}^{l+u_j}$ collected in time period t_j, where $\boldsymbol{\ell}_i$ is the location of ith reference point. Our objective is to adapt the mapping function $f^{(1)}$ to a new mapping function f, for each online time period t_j.

[1]Extension to the three-dimensional case is straight-forward.

Overall Approach

Our LeManCoR approach has the following assumptions:

1. *Physical locations* → *RSS Values*: Physical locations that are close to each other should get similar RSS values.
2. *RSS Values* → *Physical locations*: Similar RSS values at corresponding physical locations should be close to each other.
3. The *RSS values* could vary greatly over different time periods, while in the same time period they change a little.

The first two assumptions have been proven to be mostly true in indoor localization problem in wireless and sensor networks (Pan *et al.* 2006; Ferris, Fox, & Lawrence 2007), over different time periods. For example, in Figure 2, RSS S_A should be more similar to RSS S_B than RSS S_C in the old signal space, and RSS S'_A should be more similar to RSS S'_B than RSS S'_C in the new signal space, since location A is closer to location B than location C in the physical location space. The pair $\{S_A, S'_A\}$ is called *RSS corresponding pair*. The third assumption is also often true in a real world.

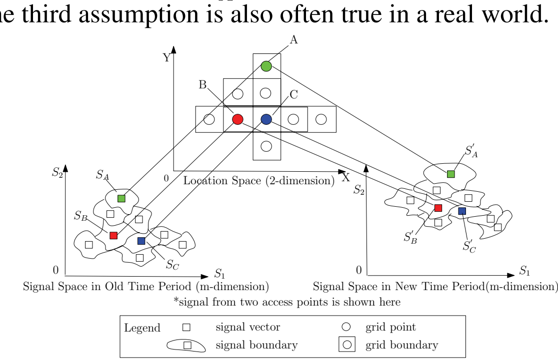


Figure 2: Correlation between location space and two different signal space

We now explain the motivation behind LeManCoR. For each online time period t_j, where $j \geq 2$, the mapping function $f^{(1)}$ learnt in time period t_1 may be grossly inaccurate in t_j. Furthermore, we only have l labeled data collected from *reference points* and u_j unlabeled data, where $l \ll l_1$ and $u_j \ll u_1$, in new time period t_j. This means in an online period t_j, we only collect a tiny amount of new data, for the propose of adapting f_1 for t_j These new data are far insufficient to rebuild a mapping function $f^{(j)}$ with high accuracy. Thus, neither $f^{(1)}$ or $f^{(j)}$, which are learnt independently, can solve the dynamic localization problem. Our motivation is that since RSS values from different time periods are multiple views of locations, we can learn a new function that can take into account both sets of data by optimizing a pair of functions in multiple views. This adapted mapping function $f'^{(j)}$ can be used in time period t_j for location estimation.

From Figure 2, we can see that this pair of functions in multiple view should agree on the same locations for each RSS corresponding pair.

Thus, the question is how to learn pairs of mapping functions together with the constraints mentioned above. In the following we will show how to solve this problem under our extended Manifold Co-Regularization framework.

Manifold Co-Regularization

Before introducing our extended Manifold Co-Regularization approach to dynamic localization problem, we first give a brief introduction to Manifold Regularization and its extension to multi-view learning, Manifold Co-Regularization.

Manifold Regularization

The standard regularization framework (Schölkopf & Smola 2002) for supervised learning solves the following minimization problem:

$$f^* = \underset{f \in H_K}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma \|f\|_K^2, \tag{1}$$

where H_k is an Reproducing Kernel Hilbert space (RKHS) of functions with kernel function K. $(x_i, y_i)_{i=1}^{l}$ is the labeled training data drawn from a probability distribution P. V is the loss function, such as squared loss for Regularized Least Squares or hinge loss function for Support Vector Machine, $\|f\|_K^2$ is a penalty term that reflects the complexity of the model. (Belkin, Niyogi, & Sindhwani 2005) extended this framework, called manifold regularization, by incorporating additional information about geometric structure of the marginal distribution P_X of P. To achieve that, (Belkin, Niyogi, & Sindhwani 2005) added an additional regularizer:

$$f^* = \underset{f \in H_K}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2, \tag{2}$$

where $\|f\|_I^2$ is an appropriate penalty term that reflects the intrinsic structure of P_X. Here γ_A controls the complexity of the function in *ambient* space while γ_I controls the complexity of the function in the *intrinsic* geometry of P_X. In (Belkin, Niyogi, & Sindhwani 2005), the third term in Equation (2) is approximated on the basis of labeled and unlabeled data using graph Laplacian associated to the data. Thus, given a set of l labeled data $\{(x_i, y_i)\}_{i=1}^{l}$ and a set of unlabeled data $\{x_j\}_{j=l+1}^{j=l+u}$, the optimization problem can be reformulated as:

$$f^* = \underset{f \in H_K}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \mathbf{f}^T L \mathbf{f},$$

where $\mathbf{f} = [f(x_1), ..., f(x_{l+u})]^T$, and L is the graph Laplacian given by $L = D - W$ where W_{ij} are the edge weights in the data adjacency graph and D is a diagonal matrix, $D_{ii} = \Sigma_{j=1}^{l+u} W_{ij}$.

Since the localization problem is a regression problem, in the rest of paper, we only focus on squared loss function $V(x_i, y_i, f) = (y_i - f(x_i))^2$.

Manifold Co-Regularization

(Sindhwani, Niyogi, & Belkin 2005) extended the manifold regularization framework, Manifold Co-Regularization, to multi-view learning. In Manifold Co-Regularization framework, we attempt to learn a function pair to correctly classify the labeled examples and to be smooth with respect to similarity structure in both views. These structures may be encoded as graphs on which regularization operators may be defined and then combined to form a multi-view regularizer.

(Sindhwani, Niyogi, & Belkin 2005) construct a multi-view regularizer by taking a convex combination $L = (1 - \alpha)L_1 + \alpha L_2$ where $\alpha \geq 0$ is a parameter that controls the influence of the two views; and L_s, where $s = 1, 2$ corresponds to the graph Laplacian matrix in each view, respectively. Thus, to learn the pair $f = (f^{(1)*}, f^{(2)*})$ is equivalent to solving the following optimization problems for $s = 1, 2$:

$$\begin{aligned} f^{(s)*} = \underset{f^{(s)} \in H_{K_s}}{\operatorname{argmin}} & \frac{1}{l} \sum_{i=1}^{l} [y_i - f^{(s)}(x_i^{(s)})]^2 + \\ & \gamma_A^{(s)} \|f^{(s)}\|_{H_{K_s}}^2 + \gamma_I^{(s)} \mathbf{f}^{(s)} L \mathbf{f}^{(s)}, \end{aligned} \tag{3}$$

where $\mathbf{f}$ denotes the vector $(f^{(s)}(x_1^{(s)}), ..., f^{(s)}(x_{l+u}^{(s)}))^T$; and the regularization parameters $\gamma_A^{(s)}, \gamma_I^{(s)}$ control the influence of unlabeled data relative to the RKHS norm. The resulting algorithm is termed Co-Laplacian RLS.

Our Extension to Manifold Co-Regularization

So far, we have reviewed how to learn a pair of classifiers under the Manifold Co-Regularization framework. However, the standard Manifold Co-Regularization approach requires all the examples (including labeled and unlabeled data) being corresponding pairs. That means the number of labeled data and unlabeled data in each view are the the same, respectively. However, in many cases, the number of examples in each view could be different, such as an indoor dynamic localization problem. Furthermore, there are only a subset of the data being corresponding pairs. For example, in our dynamic localization problem, we can collect more signal data in time period t_1 (offline phase), including l_1 labeled data and u_1 unlabeled data, while in time period t_j, we can only collect a few labeled data l from reference points and a few unlabeled data u_j, where $l \ll l_1$ and $u_j \ll u_1$. To cope with this problem, we put forward our proposed extended Manifold Co-Regularization approach to regression problem.

We assume that there are $l_1 + u_1$ data in the first view, where l_1 are labeled and u_1 are unlabeled. While there are $l_2 + u_2$ data in the second view, where l_2 are labeled and u_2 are unlabeled. l_1 and u_1 can be different from l_2 and u_2, respectively. Furthermore, there are l corresponding pairs between two views, where $l \leq min(l_1, l_2)$. Instead of Equation (3), we use the following optimization form to learn a pair of mapping functions $f = (f^{(1)*}, f^{(2)*})$:

$$(f^{(1)*}, f^{(2)*}) = \underset{f^{(1)} \in H_{K_1}, f^{(2)} \in H_{K_2}}{\operatorname{argmin}} \{$$
$$\frac{\mu}{l_1} \sum_{i=1}^{l_1} V(x_i^{(1)}, y_i^{(1)}, f^{(1)}) + \gamma_A^{(1)} \|f^{(1)}\|_{H_{K_1}}^2 + \gamma_I^{(1)} \|f^{(1)}\|_I^2 +$$
$$\frac{1}{l_2} \sum_{i=1}^{l_2} V(x_i^{(2)}, y_i^{(2)}, f^{(2)}) + \gamma_A^{(2)} \|f^{(2)}\|_{H_{K_2}}^2 + \gamma_I^{(2)} \|f^{(2)}\|_I^2 +$$
$$\frac{\gamma_I}{l} \sum_{i=1}^{l} (f^{(1)}(x_i^{(1)}) - f^{(2)}(x_i^{(2)}))^2 \}. \tag{4}$$

In Equation (4), the first three terms make the mapping function $f^{(1)}$ correctly classify the labeled examples and smooth both in function space and intrinsic geometry space. The next three terms make the same thing to the mapping function $f^{(2)}$. The last term make $f^{(1)}$ and $f^{(2)}$ agree the same labels, in our case locations, of the corresponding pairs. μ is a parameter to balance data fitting in the two views and γ_I is a parameter that regularizes the pair mapping functions.

As mentioned above, we use graph Laplacian to approximate the term $\|f\|_I$. Thus the terms $\gamma_I^{(1)} \|f^{(1)}\|_I^2$ and $\gamma_I^{(2)} \|f^{(2)}\|_I^2$ in Equation 4 can be replaced with the following two terms $\frac{\gamma_I^{(1)}}{l_1+u_1} f^{(1)T} L_1 f^{(1)}$ and $\frac{\gamma_I^{(2)}}{l_2+u_2} f^{(2)T} L_2 f^{(2)}$, respectively.

Using $f^{(1)*} = \sum_{i=1}^{l_1+u_1} \alpha_i^* K_1(x^{(1)}, x_i^{(1)})$ and $f^{(2)*} = \sum_{i=1}^{l_2+u_2} \beta_i^* K_2(x^{(2)}, x_i^{(2)})$ to substitute $f^{(1)*}$ and $f^{(2)*}$ in Equation 4, we arrive at the following objective function of the (l_1+u_1)-dimensional variable $\alpha = [\alpha_1, \cdots, \alpha_{l_1+u_1}]$ and (l_2+u_2)-dimensional variable $\beta = [\beta_1, \cdots, \beta_{l_2+u_2}]$:

$$\alpha^*, \beta^* = \underset{\alpha \in R^{l_1+u_1}, \beta \in R^{l_2+u_2}}{\operatorname{argmin}} \{$$
$$\frac{\mu}{l_1} (Y_1 - J_1 K_1 \alpha)^T (Y_1 - J_1 K_1 \alpha) + \gamma_A^{(1)} \alpha^T K_1 \alpha +$$
$$\frac{\gamma_I^{(1)}}{l_1+u_1} \alpha^T K_1 L_1 K_1 \alpha + \frac{1}{l_2} (Y_2 - J_2 K_2 \beta)^T (Y_2 - J_2 K_2 \beta) +$$
$$\gamma_A^{(2)} \beta^T K_2 \beta + \frac{\gamma_I^{(2)}}{l_2+u_2} \beta^T K_2 L_2 K_2 \beta +$$
$$\frac{\gamma_I}{l} (J_1' K_1 \alpha - J_2' K_2 \beta)^T (J_1' K_1 \alpha - J_2' K_2 \beta) \}, \tag{5}$$

where K_s is the kernel matrix over labeled and unlabeled examples in each view. Y_s is an $(l_s + u_s)$ dimensional label vector given by: $Y = [y_1, ..., y_{l_s}, 0, ..., 0]$, where the first l_s examples are labeled while the others are unlabeled. In our case, Y_s corresponds to a vector of locations of each coordinate. J_s is a diagonal matrix given by $J_s(i,i) = |Y_s(i)|$. J_s': is a $(l \times (u_s + l_s))$ given by $J_s'(i,i) = 1$ for $i = 1, 2, ..., l$, otherwise, $J_s'(i,j) = 0$. l is the number of corresponding pairs. In the above formulas, $s = 1$ and 2, respectively.

The derivatives of the objective function, respectively, vanish at the pair of minimizers α and β:

$$\begin{aligned}(\tfrac{\mu}{l_1} J_1 K_1 + \gamma_A^{(1)} I_1 + \tfrac{\gamma_I}{l} J_1'^T J_1' K_1 + \tfrac{\gamma_I^{(1)}}{l_1+u_1} L_1 K_1)\alpha \\ -\tfrac{\gamma_I}{l} J_1'^T J_2' K_2 \beta = \tfrac{\mu}{l_1} Y_1,\end{aligned} \tag{6}$$

$$\begin{aligned}(\tfrac{1}{l_2} J_2 K_2 + \gamma_A^{(2)} I_2 + \tfrac{\gamma_I}{l} J_2'^T J_2' K_2 + \tfrac{\gamma_I^{(2)}}{l_2+u_2} L_1 K_1)\beta \\ -\tfrac{\gamma_I}{l} J_2'^T J_1' K_1 \alpha = \tfrac{1}{l_2} Y_2,\end{aligned} \tag{7}$$

which leads to the following solution.

$$\alpha^* = (A_1 - B_1 B_2^{-1} A_2)^{-1} (\tfrac{\mu}{l_1} Y_1 + \tfrac{1}{l_2} B_1 B_2^{-1} Y_2), \tag{8}$$

$$\beta^* = (B_2 - A_2 A_1^{-1} B_1)^{-1} (\tfrac{\mu}{l_1} A_2 A_1^{-1} Y_1 + \tfrac{1}{l_2} Y_2), \tag{9}$$

where

$B_1 = \frac{\gamma_I}{l} J_1'^T J_2' K_2; \quad A_2 = \frac{\gamma_I}{l} J_2'^T J_1' K_1;$

$A_1 = \frac{\mu}{l_1} J_1 K_1 + \gamma_A^{(1)} I_1 + \frac{\gamma_I}{l} J_1'^T J_1' K_1 + \frac{\gamma_I^{(1)}}{l_1+u_1} L_1 K_1;$

$B_2 = \frac{1}{l_2} J_2 K_2 + \gamma_A^{(2)} I_2 + \frac{\gamma_I}{l} J_2'^T J_2' K_2 + \frac{\gamma_I^{(2)}}{l_2+u_2} L_1 K_1.$

We call this resulting algorithm extended Co-Manifold RLS (eManCoR). eManCoR extends the Co-LapRLS to the situation that the number of training data in multi-view is different and there are only a subset of the data having pairwise correspondence among different views. Thus, we can apply eManCoR algorithm to learn a pair of mapping functions in dynamic localization problem. The localization version of eManCoR is called LeManCoR

The LeManCoR Algorithm

Our dynamic location-estimation algorithm LeManCoR, which is based on our Extension to Manifold Co-Regularization above, has three phases: an offline training phase, an online adaptation phase and an online localization phase.

• **Offline Training Phase**

1. Collect l_1 labeled signal-location pairs $\{(\mathbf{s}_i^{(1)}, \boldsymbol{\ell}_i^{(1)})\}_{i=1}^{l_1}$ at various locations (the first l pairs are collected from reference points) and u_1 unlabeled signal examples $\{\mathbf{s}_j^{(1)}\}_{j=l_1+1}^{l_1+u_1}$.
2. For each signal vector $\mathbf{s}_i^{(1)} \in S^{(1)}$, build an edge to its k nearest neighbors in Euclidean distance measure. After that, the adjacency graph and weight matrix can be used to form the the graph Laplacian L_1.
3. Apply LapRLS to learn the mapping function $f_x^{(1)}$ and $f_y^{(1)}$, which are for x and y coordinates, respectively.

• **Online Adaptation Phase**

1. For each time period t_j, $j \geq 1$:
   If $j = 1$, the mapping function $f_x^{(1)}$ and $f_y^{(1)}$ can be applied to estimate the location directly, $f_x'^{(j)} = f_x^{(1)}$ and $f_y'^{(j)} = f_y^{(1)}$.
   If $j > 1$, randomly collect a few unlabel data u_j around the environment and get l labeled data: $\{(\mathbf{s}_i^{(j)}, \boldsymbol{\ell}_i)\}_{i=1}^{l}$, where $\boldsymbol{\ell}_i$ is the location of ith reference point, from reference points. Thus, each pair $\{(\mathbf{s}_i^{(1)}, \mathbf{s}_i^{(j)})\}$ is a corresponding pair, where $i = 1, ..., l$.
2. Apply eManCoR to learn the pair of mapping functions $\{f_x^{*(1)}, f_x^{*(j)}\}$ and $\{f_y^{*(1)}, f_y^{*(j)}\}$. We use $f_x^{*(1)}$ and $f_y^{*(1)}$ as the new mapping functions in time period t_j instead of

$f_x^{*(j)}$ and $f_y^{*(j)}$. This is because the data in t_j are insufficient to build a strong classifier. The learning process of $f_x^{*(j)}$ and $f_y^{*(j)}$ is to adapt the parameter of $f_x^{*(1)}$ and $f_y^{*(1)}$ with new data in t_j. Thus $f_x'^{(j)} = f_x^{*(1)}$ and $f_y'^{(j)} = f_y^{*(1)}$.

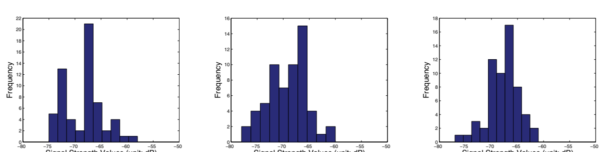
- **Online Localization Phase**

1. For each time period t_j, each signal vector $\tilde{\mathbf{s}}$ collected by a *mobile* node, $\tilde{\boldsymbol{\ell}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = (f_x'^{(j)}(\tilde{\mathbf{s}}), f_y'^{(j)}(\tilde{\mathbf{s}}))$ is the location estimation.
2. Optionally, we can apply a Bayes Filter (Fox *et al.* 2003; Hu & Evans 2004) to smooth the trajectory and enhance the performance of localization.

Experimental Results

We evaluate the performance of LeManCoR in an office environment in an academic building, which is about $30 \times 40m^2$, over six time periods: *12:30am-1:30am*, *08:30am-09:30am*, *12:30pm-1:30pm*, *04:30pm-05:30pm*, *08:30pm-09:30pm* and *10:30pm-11:30pm*. The distributions of RSS values of three different time periods shown in Figure 3. For every time period, we collect totally 1635 WiFi examples in 81 grids, each of which is about $1.5 \times 1.5m^2$. Thus we obtained six data sets of WiFi signal data. We install several USB Wireless network adapters on various PCs around the environment. These adapters act as reference points to collect the labeled signal data over different time periods. We consider the midnight time period *12:30am-1:30am* as the offline training phase, when we randomly choose 70% of the 1635 examples as the training data, of which only 30% have labels, that means the corresponding locations are known. The other five periods are considered as online phase. For each of them, we randomly choose 20% of examples as unlabeled data used in the adaptation phase.

We use a Gaussian kernel $exp(-\|x_i - x_j\|)^2/2\sigma^2)$ for Equations (6) and (7), which is widely used in localization problems for adapting the noisy characteristic of radio signal (Nguyen, Jordan, & Sinopoli 2005), where σ is set to 0.5. For $\gamma_A^{(1)}$, $\gamma_A^{(2)}$, $\gamma_I^{(1)}$, $\gamma_I^{(2)}$, μ and $\gamma_I$in Equations (6) and (7), we set $\frac{\gamma_A^{(1)} l_1}{l_1+u_1} = \frac{\gamma_A^{(2)} l_2}{l_2+u_2} = 0.05$ and $\gamma_I^{(1)} l_1 = \gamma_I^{(2)} l_2 = 0.045$. $\mu = 1$ and $\frac{l_1 \gamma_I}{l} = 0.3$, where $l = l_2$ in our case.



(a) 12:30-1:30 (am) (b) 8:30-9:30 (am) (c) 12:30-1:30 (pm)

Figure 3: Variations of signal-strength histograms over different time periods at the same location

Dynamic Location Estimation Accuracy

For evaluating the adaptability of LeManCoR, we compare it with a static mapping method LeMan (Pan *et al.* 2006). Since LeMan is also based on manifold setting, we can see the adaptability of LeManCoR clearly by the comparison. We first show the comparison results in Figure 4 (see next page) where the number of reference points is fixed to 10, and we further show the impact of different number of reference points to LeManCoR in Figure 5. In Figure 4, *LeManCoR* denotes our model while *LeMan* denotes the static mapping function leant in time period *12:30am-01:30am* by LapRLS. *LeMan2* gives that for each online time period, reconstructed mapping function using the up-to-date reading of reference points and a few new unlabeled data by LapRLS. *LeMan3* denotes constructing the mapping function from RSS data collected in *12:30am-01:30am*, up-to-date reading of reference points and a few unlabeled data. Figure 4(a) shows that when the training data and test data are in the same time period, *12:30am-01:30am*, *LeMan* can get a high location estimation accuracy. However, when this static mapping function is applied to different time periods, the accuracy decreases fast. See Figure 4(b), 4(c), 4(d), 4(e), 4(f). The curves of *LeMan2* and show that only several labeled and a few unlabeled data are far insufficient to build a new classifier. In the contrary, our LeManCoR system can adapt the mapping function effectively by only 10 reference points and 300 unlabeled examples in new time period.

Robustness to the Number of Reference Points

In this experiment, we study how is the impact of the number of reference points to LeManCoR. We set the number of reference points from 5 to 20, and compare the accuracy with error distance 3.0m. The result is shown in Figure 5. As can be seen, LeManCoR is robust to the number of reference points.

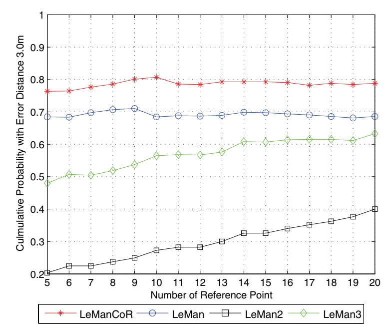
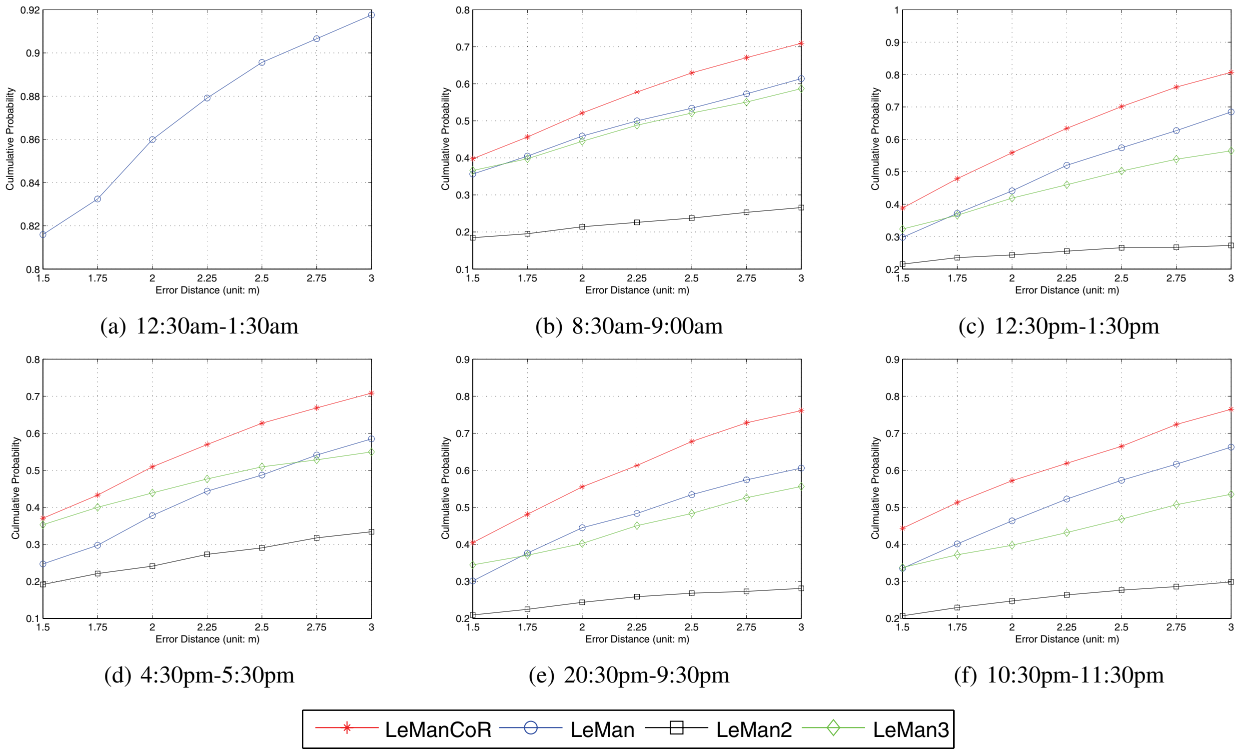


Figure 5: Varying the number of reference points in 12:30pm-1:30pm, where accuracy is with error distance 3m

Due to space limitation, several other issues are not discussed. For example we also varied the number of unlabeled examples in online time periods, and observed that our LeManCoR is also robust to the number of unlabeled data using in online adaptation phase. We also test the running time of LeManCoR on our Pentium 4 PC. For each time period, learning an adapted mapping function from 800 old signal examples and 300 new signal examples only took 10 seconds.

Conclusions and Future Work

In this paper, we describe LeManCoR, a dynamic localization approach which transfers geometric knowledge of the WiFi data in different time periods by modified manifold co-regularization. Our model is based on the observations that in each different time period, similar signals strength imply close locations, and the pairwise signals over different



(a) 12:30am-1:30am
(b) 8:30am-9:00am
(c) 12:30pm-1:30pm
(d) 4:30pm-5:30pm
(e) 20:30pm-9:30pm
(f) 10:30pm-11:30pm

Figure 4: Comparison of accuracy over different time periods when the number of reference points is 10

time periods should correspond to the same locations. The mapping function between the signal space and the physical location space is adapted dynamically by several labeled data from reference points and a few unlabeled data in a new time period. Experimental results show that our model can adapt the static mapping function effectively and is robust to the number of reference points. In the future, we will extend LeManCoR to continuous time instead of discrete time periods.

References

Bahl, P.; Balachandran, A.; and Padmanabhan, V. 2000. Enhancements to the RADAR user location and tracking system. Technical report, Microsoft Research.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2005. On Manifold Regularization. In Robert G. Cowell and Zoubin Ghahramani., ed., *Proceedings of AISTATS 2005*.

Ferris, B.; Fox, D.; and Lawrence, N. 2007. WiFi-SLAM Using Gaussian Process Latent Variable Models. In *Proceedings of IJCAI 2007*.

Ferris, B.; Haehnel, D.; and Fox, D. 2006. Gaussian Processes for Signal Strength-Based Location Estimation. In *Proceedings of Robotics: Science and Systems 2006*.

Fox, D.; Hightower, J.; Liao, L.; Schulz, D.; and Borriello, G. 2003. Bayesian filtering for location estimation. *IEEE Pervasive Computing* 2(3):24–33.

Haeberlen, A.; Flannery, E.; Ladd, A. M.; Rudys, A.; Wallach, D. S.; and Kavraki, L. E. 2004. Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In *Proceedings of MobiCom 2004*.

Hu, L., and Evans, D. 2004. Localization for mobile sensor networks. In *Proceedings of MobiCom 2004*.

Krishnan, P.; Krishnakumar, A. S.; Ju, W.-H.; Mallows, C.; and Ganu, S. 2004. A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. In *Proceedings of INFOCOM 2004*.

LaMarca, A.; Hightower, J.; Smith, I.; and Consolvo, S. 2005. Self-Mapping in 802.11 Location Systems. In *Proceedings of Ubicomp 2005*.

Nguyen, X.; Jordan, M. I.; and Sinopoli, B. 2005. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks* 1(1):134–152.

Pan, J. J.; Yang, Q.; Chang, H.; and Yeung, D.-Y. 2006. A Manifold Regularization Approach to Calibration Reduction for Sensor-Network Based Tracking. In *Proceedings of AAAI 2006*.

Schölkopf, B., and Smola, A., eds. 2002. *Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, Massachusetts: MIT Press.

Sindhwani, V.; Niyogi, P.; and Belkin, M. 2005. A Co-Regularization Approach to Semi-supervised Learning with Multiple Views. In *Workshop on Learning with Multiple Views at ICML 2005*.

Yin, J.; Yang, Q.; and Ni, L. 2005. Adaptive temporal radio maps for indoor location estimation. In *Proceedings of PerCom 2005*.