Adaptive Localization Through Transfer Learning in Indoor Wi-Fi Environment

Zhuo Sun, Yiqiang Chen, Juan Qi and Junfa Liu
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{sunzhuo, yqchen, qijuan, liujunfa}@ict.ac.cn

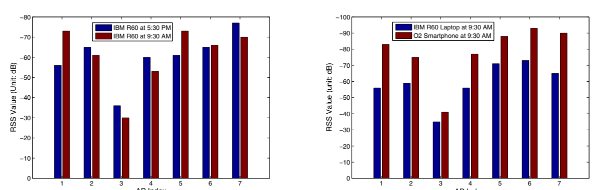
Abstract

In a Wi-Fi based indoor localization system (WILS), mobile clients use received Wi-Fi signal strength to determine their locations. A major problem is the variation of signal distributions caused by multiple factors, which makes the old localization model inaccurate. Therefore, the transfer learning problem in a WILS aims to transfer the knowledge from an old model to a new one. In this paper, we study the characteristics of signal variation and conclude the chief factors as time and devices. An algorithm LuMA is proposed to handle the transfer learning problem caused by these two factors. LuMA is a dimensionality reduction method, which learns a mapping between a source data set and a target data set in a low-dimensional space. Then the knowledge can be transferred from source data to target data using the mapping relationship. We implement a WILS in our wireless environment and apply LuMA on it. The online performance evaluation shows that our algorithm not only achieves better accuracy than the baselines, but also has ability for adaptive localization, regardless of time or device factors. As a result, the calibration efforts on new training data can be greatly reduced.

1 Introduction

Although global positioning system (GPS) and wireless E911 services address the issue of location finding, these technologies can not provide an accurate indoor location because of their technical constraints. Considering the expanding coverage of Wi-Fi signals, indoor localization usually uses existing wireless local access network (WLAN) infrastructures for positioning. In a Wi-Fi based indoor localization system (WILS), mobile clients receive Wi-Fi signals from different access points (APs) at each position and the received signal strength (RSS) values can be combined as a vector called a fingerprint, indicating the position information. Therefore, the most common approach for indoor localization is to collect RSS at different positions and use machine learning methods to do pattern matching. This approach was first brought out in RADAR system [1] and adopted by later WILSs like PlaceLab [5], Horus [11] etc.

As it is necessary to collect enough calibrated RSS fingerprints (i.e. labeled data), human needs to walk through the environment with a mobile client to collect the RSS values and mark down the geographic location. It would cost tremendous human efforts in a large-scale environment. While in an actual indoor Wi-Fi environment, RSS values are very noisy and fluctuant due to multi-path and shadow fading effects. Superadded with the affection to signal propagation caused by temperature, humidity and movement of people, the distributions of RSS values of different time periods vary a lot. Moreover, various mobile clients with different types of wireless cards have apparently distinct capacities of sensing Wi-Fi signals, some are especially sensitive but some are not. Therefore, the distributions of RSS values collected by different clients also vary with each other. Figure 1 shows the discrepancy of signal distribution caused by these two main factors. One client collected dissimilar RSS values at different time periods of a day in Fig.1(a), and two types of clients collected disparate RSS values at a same moment in Fig.1(b).



(a) Signals detected by an IBM R60 at different time periods.

(b) Signals detected by an IBM R60 and an O2 smartphone at a same time period.

Figure 1. Signal distribution discrepancy caused by time and device factors.

When the signal distribution changes, new RSS fingerprints should be collected again in order to keep the localization accuracy, which is costly or infeasible sometimes. In machine learning, transfer learning aims to solve the problem that the training data in a source domain and the test

978-0-7695-3495-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ICMLA.2008.53

data in a target domain follow different distributions [6]. As a consequence, using transfer learning algorithms to reduce the repeated calibration efforts makes a WILS much more adaptive and practical. Although several transfer learning algorithms have been proposed such as [7], [13], [12], none of them consider time and device factors together. Accordingly, no WILSs using machine learning methods are able to accommodate complicated situations adaptively indeed.

In this paper, we propose a transfer learning algorithm based on manifolds alignment to solve the disparate signal distributions caused by both time and devices. Manifolds alignment is a dimensionality reduction method, which learns mappings between different data sets that are characterized by the same underlying manifold and discovers the corresponding relationship in a low-dimensional space. Then the correlation is used to transfer the knowledge from source data to target data. We implement a real-world WILS to apply our algorithm LuMA. The performance is evaluated compared with some baselines and the experimental results show that our algorithm outperforms the baseline models for solving the practical transfer learning problem in a WILS effectively. This paper makes the following two main contributions. First, we propose a novel solution for the pragmatic transfer learning problem caused by both time and device factors. Second, our algorithm performs well with only a small amount of labeled target data and greatly reduces the calibration effort.

2 Related Work

Previous WILSs such as PlaceLab [5], Horus [11] assume that the Wi-Fi signal distributions do not change a lot over time and their work only concentrate on one type of clients, consequently the transfer learning problem is not considered at all.

Recently, transfer learning in indoor localization domain has attracted much attention as it is closely related to practicality of WILSs. Several transfer learning algorithms have been proposed to accommodate the dynamic indoor environment. [10] assumes that the RSS value at each location can be calculated as a linear aggregate of the RSS values of neighboring benchmarks for each access point and this linear relationship remains the same over different time periods. Then at each location a model tree is built up for each access point. [13] transfers the HMM for a new time period. It uses the neighboring relationship as a bridge to transfer the knowledge and treats the trace as a Markov chain, then a HMM for the new time period is trained out. [12] considers the transfer learning problem over devices and treats multiple devices as multiple learning tasks. Then it learns the classifier in a latent feature space. However, these algorithms consider transfer learning over either time or devices and none of them are applied in a real-world WILS.

3 Methodology

3.1 Problem Statement

Suppose there are totally m APs deployed around a wireless environment, which is divided into n grids. At each grid, a mobile client taken by a user can receive Wi-Fi signals from some APs periodically. The RSS values can be defined as a signal vector $\mathbf{s} = (s_1, s_2, ..., s_m) \in \mathbb{R}^m$, where s_i stands for the RSS value received from AP$_i$ and we fill the missing values with -100, the lowest signal strength that can be detected. The signal vector with its position label p forms the labeled data $\{(\mathbf{s}, p)\}$, and the unlabeled data is defined as $\{(\mathbf{s})\}$.

Consider the data collected at two different time periods T_0 and T_t. There are enough labeled data $\{(\mathbf{s}_i^{(0)}, p_i)\}_{i=1}^{l_0}$ collected at T_0, but we wish to use these data for localization at T_t, when only a small amount of labeled data $\{(\mathbf{s}_i^{(t)}, p_i)\}_{i=1}^{l_t}, l_t \ll l_0$ at T_t are available. Optionally, some unlabeled data $\{(\mathbf{s}_i^{(t)}\}_{i=1}^{u_t}$ at T_t are easy to get for assistance.

The data collected by two different devices V_{src} and V_{tar} have a similar description like above, that enough labeled data $\{(\mathbf{s}_i^{(src)}, p_i)\}_{i=1}^{l_{src}}$ are collected by V_{src} while only limited labeled data $\{(\mathbf{s}_i^{(tar)}, p_i)\}_{i=1}^{l_{tar}}, l_{tar} \ll l_{src}$ are collected by V_{tar}. We hope to predict the location of V_{tar} using mainly the data collected by V_{src} with the help of $\{(\mathbf{s}_i^{(tar)}, p_i)\}_{i=1}^{l_{tar}}$ and some unlabeled data $\{(\mathbf{s}_i^{(tar)})\}_{i=1}^{u_{tar}}$.

Generally, these two situations in a WILS can be formulated as a uniform transfer learning problem. Given enough labeled data $\mathscr{D}_{src}^l$ of a source domain $\mathcal{S}$, a prediction of a target domain $\mathcal{T}$ can be made out with a few labeled data $\mathscr{D}_{tar}^l$ as benchmarks and some optional unlabeled data $\mathscr{D}_{tar}^u$.

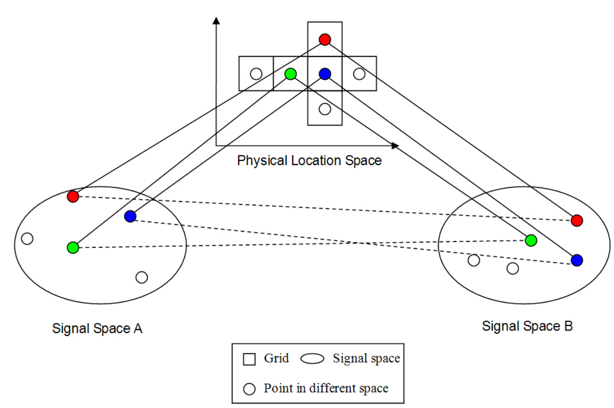


Figure 2. Correlation between two different signal spaces and physical space

The fundamental of transfer learning is based on some observations about the characteristics of Wi-Fi signals. First and the most important, the nearby locations have more similar RSS values than those that are far away. Second, although different distributions of signals caused by different time periods or devices form different signal spaces, they are based on a common physical space, which makes the transfer learning between them feasible. Figure 2 illustrates this corresponding relationship. The points with the same color in different signal spaces correspond to each other through their common physical locations (shown as broken lines). The question is, given a part of corresponding pairwise points, such as the colored points in Fig.2, how can we learn out the mappings of the left uncolored ones?

3.2 Manifolds Alignment by Pairwise Correspondence

Traditional dimensionality reduction methods, like linear methods such as PCA (principal components analysis) [4], graph based methods such as LLE (locally linear embedding) [8], ISOMAP (isometric feature mapping) [9] etc, concentrate on only one data set without considering the relationship of different data sets. However, different data sets could also have latent common features in a low-dimensional space that may be useful or have physical meanings. Here we introduce the method of manifolds alignment [3], which is a dimensionality reduction method based on manifolds with given constraints. It can easily align the low-dimensional representation of the data sets given some pairwise correspondence between the data sets. We make some extension on this method to transfer the knowledge from a source domain to a target domain.

It has been proposed in [2] that manifold learning can successfully learn a low-dimensional embedding by constructing a weighted graph that captures local structure in the data. Suppose $X = \{\mathrm{x}_1, ..., \mathrm{x}_m\} \subset \mathbb{R}^{D_h}$ is a data set in a high-dimensional space, a weighted matrix W is constructed where $W_{ij} \neq 0$ when the ith and jth data points in X are neighbors ($i \sim j$), otherwise $W_{ij} = 0$. The generalized graph Laplacian L is then defined as:

$$L_{ij} := \begin{cases} \sum_{j \sim i} W_{ij} & \text{if } i = j \\ -W_{ij} & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

If the graph is connected, L will have a single zero eigenvalue associated with the uniform vector $\boldsymbol{e} = [1, 1, ..., 1]^T$.

A low-dimensional (D_l) embedding of the data can be computed from the graph Laplacian in the following manner. Define a real valued function $f : D_h \to D_l$ on the graph, to find an optimal embedding addresses to the following optimization problem:

$$\arg\min_{\boldsymbol{f}} \quad \boldsymbol{f}^T L \boldsymbol{f} = \tfrac{1}{2} \sum_{i,j} (f_i - f_j)^2 W_{ij} \qquad (2)$$

$$\text{s.t.} \qquad \boldsymbol{f}^T \boldsymbol{f} = 1 \text{ and } \boldsymbol{f}^T \boldsymbol{e} = 0$$

The optimal solutions $\boldsymbol{f}$ are the eigenvectors of L with the smallest D_l non-zero eigenvalues, creating a low-dimensional embedding with a similar data structure as the original high-dimensional data.

Now consider constructing the low-dimensional embeddings of two related data sets simultaneously. Suppose X, Y are two different data sets of high dimension, which have subsets X_p and Y_p in pairwise correspondence (e.g. the same labels), while the other data are denoted by X_s and Y_s, it is possible to match X_s and Y_s using an aligned manifold embedding.

Let $\boldsymbol{f}$ and $\boldsymbol{g}$ denote real-valued functions defined on the respective graphs of X and Y, and the pairwise correspondences are indicated by the labels $x_i \leftrightarrow y_i, i \in p$. According to (2), $\boldsymbol{f}$ and $\boldsymbol{g}$ represent coordinates of the optimal low-dimensional embeddings that are extracted from each data set separately. As the pairwise points in two data sets should align well, the values of $\boldsymbol{f}$ and $\boldsymbol{g}$ for these pairs should be similar. Generalizing the single graph embedding algorithm, optimal corresponding embedding is the following optimization problem:

$$C(\boldsymbol{f}, \boldsymbol{g}) = \mu \sum_{i \in p} |f_i - g_i|^2 + \boldsymbol{f}^T L^x \boldsymbol{f} + \boldsymbol{g}^T L^y \boldsymbol{g} \qquad (3)$$

where L^x and L^y are the graph Laplacian matrices. The first term penalizes the discrepancies between $\boldsymbol{f}$ and $\boldsymbol{g}$ on the pairwise points, and the latter terms preserve the smoothness of manifolds in low dimension. However, this optimization does not consider whether the data set sizes of X and Y are comparable. Instead, we extend it by adding some adjusting parameters:

$$C(\boldsymbol{f}, \boldsymbol{g}) = \mu \sum_{i \in p} |f_i - g_i|^2 + \lambda_1 \boldsymbol{f}^T L^x \boldsymbol{f} + \lambda_2 \boldsymbol{g}^T L^y \boldsymbol{g} \qquad (4)$$

The optimization in (4) is ill-defined because it is not invariant to simultaneous scaling of $\boldsymbol{f}$ and $\boldsymbol{g}$. Therefore, we define $\boldsymbol{h} = [\boldsymbol{f}^T \boldsymbol{g}^T]^T$ and to minimize (4) is equivalent to minimize the Rayleigh quotient:

$$\arg\min_{\boldsymbol{h}} \quad \widetilde{C}(\boldsymbol{h}) = \frac{\boldsymbol{h}^T L^z \boldsymbol{h}}{\boldsymbol{h}^T \boldsymbol{h}}, \quad \text{s.t.} \quad \boldsymbol{h}^T \boldsymbol{e} = 0, \qquad (5)$$

where L^z is defined as

$$L^z = \begin{bmatrix} \lambda_1 L^x + U^x & -U^{xy} \\ -U^{yx} & \lambda_2 L^y + U^y \end{bmatrix}, \qquad (6)$$

and U^x,U^y,U^{xy} and U^{yx} are matrices that have non-zero elements only on the diagonal

$$U_{ij} = \begin{cases} \mu, & i = j \in p \\ 0, & \text{otherwise} \end{cases}$$

Furthermore, the coefficient μ in (3) weights the importance of the correspondence term relative to the smoothness terms. In the limit $\mu \to \infty$, the result is equivalent to imposing hard constraints $f_i = g_i$ for $i \in p$. Then the optimization is given by the eigenvalue problem:

$$\begin{aligned} \arg\min_{\boldsymbol{h}} \quad & \widetilde{C}(\boldsymbol{h}) := \frac{\boldsymbol{h}^T L^z \boldsymbol{h}}{\boldsymbol{h}^T \boldsymbol{h}} \\ \text{s.t.} \quad & \boldsymbol{h}^T \boldsymbol{e} = 0 \end{aligned} \tag{7}$$

where $\boldsymbol{h}$ and L^z are defined as

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f}_p = \boldsymbol{g}_p \\ \boldsymbol{f}_s \\ \boldsymbol{g}_s \end{bmatrix}, \tag{8}$$

$$L^z = \begin{bmatrix} \lambda_1 L^x_{pp} + \lambda_2 L^y_{pp} & \lambda_1 L^x_{ps} & \lambda_2 L^y_{ps} \\ \lambda_1 L^x_{sp} & \lambda_1 L^x_{ss} & 0 \\ \lambda_2 L^y_{sp} & 0 & \lambda_2 L^y_{ss} \end{bmatrix} \tag{9}$$

The optimization is solved by finding the smallest r-nonzero eigenvalues of L^z without an exact parameter μ. Then the r-dimensional embedding is obtained by the r-nonzero eigenvectors of L^z. From this common embedding, a point in one data set can be mapped to the corresponding point in the other data set using nearest neighbors, without inferring a direct transformation between the two data sets.

4 Algorithm

As stated in Section 3.1, the transfer learning problem in a WILS addresses to using the labeled data $\mathscr{D}^l_{src}$ in a source domain to predict the test data in a target domain, some labeled data $\mathscr{D}^l_{tar}$ are given as benchmarks and unlabeled data $\mathscr{D}^u_{tar}$ are optional. We propose our algorithm LuMA, localization using manifolds alignment, to solve the transfer learning problem caused by both time and device factors. The basic idea behind LuMA is to utilize the correlation between the $\mathscr{D}^l_{src}$ and $\mathscr{D}^l_{tar}$ to find out the corresponding relationship of $\mathscr{D}^u_{tar}$ with $\mathscr{D}^l_{src}$, therefore, the amount of labeled data in the target domain can be enlarged and new localization model is easily trained out.

Algorithm: Localization using Manifolds Alignment

Input: Labeled data $\mathscr{D}^l_{src} = \{(\mathbf{s}^{(src)}_i, p_i)\}^{l_{src}}_{i=1}$ in a source domain, labeled data $\mathscr{D}^l_{tar} = \{(\mathbf{s}^{(tar)}_i, p_i)\}^{l_{tar}}_{i=1}$ and unlabeled data $\mathscr{D}^u_{tar} = \{(\mathbf{s}^{(tar)}_i)\}^{u_{tar}}_{i=1}$ in a target domain

Output: Localization model in the target domain

Steps:

1. Find out l_{tar} corresponding data in $\mathscr{D}^l_{src}$ to pair $\mathscr{D}^l_{tar}$, like X_p and Y_p in Section 3.2, make sure the sequences are in a corresponding order. The remaining $l_{src} - l_{tar}$ data in $\mathscr{D}^l_{src}$ is X_s and $\mathscr{D}^u_{tar}$ is Y_s.
2. Combine X_p and X_s as X and compute the graph Laplacian L^x of X according to (1). The same approach is repeated on Y.
3. Compute the joint graph Laplacian L^z according to (9).
4. Calculate the eigenvalues of L^z, then r eigenvectors with the smallest r-nonzero eigenvalues construct a aligned embedding.
5. For each data in $\boldsymbol{g}_s$, its label is assigned by its nearest neighbor in $\boldsymbol{f}_s$. This alignment results in a labeled $\mathscr{D}^u_{tar}$.
6. Use $\mathscr{D}^l_{tar}$ and labeled $\mathscr{D}^u_{tar}$ together, a localization model in the target domain can be trained out by supervised algorithms such as Support Vector Machine (SVM), k-Nearest Neighbor (kNN) and so on.

The adjusting parameters λ_1 and λ_2 are determined according to the size of data sets, and we find when $\lambda_1 = \frac{l_{tar}+u_{tar}}{l_{src}+l_{tar}+u_{tar}}$, $\lambda_2 = \frac{l_{src}}{l_{src}+l_{tar}+u_{tar}}$ the alignment has the best result. As our method reduces dimension based on eigenvalue problem, the dimension r can be decided like PCA that extracts principle components by choosing larger eigenvalues, thereby we set it to choose more than 95% of the sum. We also discover that simple kNN algorithm performs quite well in localization problem, so we adopt kNN as a supervised learning methods for Step 6. Moreover, since RSS values are noisy, instead of the familiar Euclidean distance metric (2-norm) in kNN, we use Manhattan distance metric (1-norm), which puts more importance on presence or absence of signals than the amount of change and increases the accuracy. After the localization model in the target domain is trained out, a mobile client collects online RSS values as inputs of the model for determining its real-time location.

5 Experiments

5.1 System Setup

In order to evaluate the performance of LuMA in a real-world WILS, we establish our own wireless network environment and build up the whole system. Our experimental test-bed is deployed on the 3rd floor of our academic building with an area of about $30m \times 15m$, covering a hallway and five rooms. We deploy 7 TENDA APs around the area to set up an IEEE 802.11b wireless network infrastructure. These APs are denoted by red triangles in Fig.3 and the whole area is divided into 161 grids for signal collection, each with a size of $1m \times 1m$.

The whole system is built up based on this experimental test-bed, including a DELL PC equipped with a TP-Link

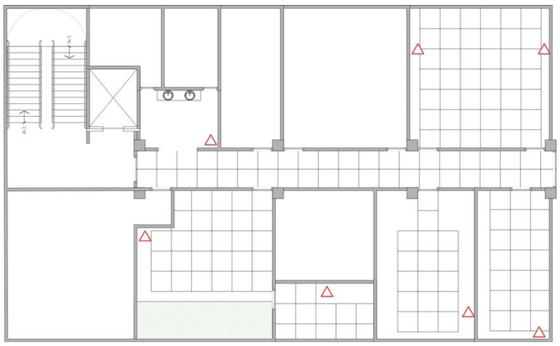


Figure 3. The layout of the experimental testbed in our building

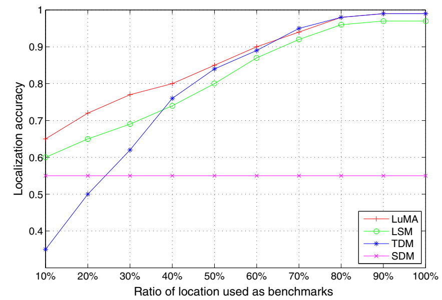
wireless card served as a localization server and two types of mobile clients: an IBM R60 laptop and an O2 Xda Atom Life smartphone. We use the IBM R60 laptop to collect RSS data grid by grid one-off as the source data at 9:30 AM. Then the source data will be combined with some target data to train the localization model of target domain. As the target data is different due to multiple factors, here we first consider the time factor and device factor separately, then both of them together.

5.2 Performance Evaluation

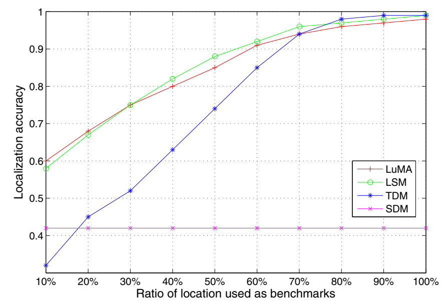
To test the transfer learning performance of LuMA over different time periods, we use the IBM R60 laptop again to collect some labeled data as benchmarks at 5:30 PM (i.e. the target domain). Since the labeled data need manual calibration, the amount is limited. A few unlabeled data can be easily got when we take the client and walk through the environment. After these target data are collected, LuMA is run on the localization server to train the new model for this time period. Then we use it to test the localization accuracy online. We test 100 positions each time for 10 times in all, and the average accuracy (within an error distance of $3m$ [1]) is shown in Fig.4(a), compared with three models as baselines: trained only by source data (SDM), trained only by target data (TDM) and using a linear shift algorithm (LSM).

It can be observed that SDM gets the worst result, indicating that transfer learning is necessary in a real-world WILS. LSM assumes that the source data have linear relationships with the target data, that is, at each AP, the RSS value of source data $s_{src} = \lambda s_{tar} + \Delta s$. However, the signal change over time is not a simple linear relationship, so LSM has a worse performance than LuMA. TDM first performs

[1]Error distance is calculated by the Euclidean distance between a predicted location and its ground truth value. The predicted location within the error distance is counted as an accurate prediction.



(a) Signal distributions are affected by time.



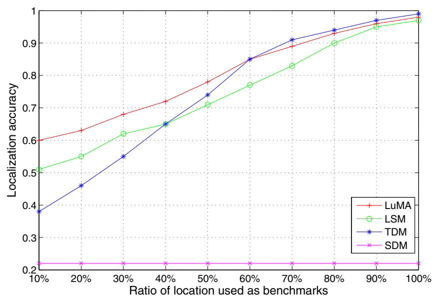
(b) Signal distributions are affected by devices.

Figure 4. Localization accuracy with different affections.

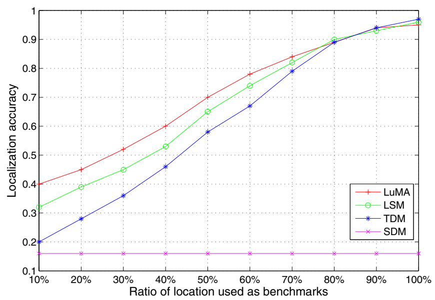
not very well, but the accuracy increases rapidly along with the increase of benchmarks. In practice, it would cause a lot of efforts to get more than 50% benchmarks. Therefore, our algorithm LuMA performs well when given only a small amount of benchmarks on the transfer learning problem over time periods.

To test the transfer learning performance over different devices, we remain the source data collected by IBM R60 laptop at 9:30 AM, but the target now is an O2 Xda Atom smartphone. We use the smartphone to collect some target data also at 9:30 AM and localization is tested on the smartphone at 100 positions each time for 10 times. A similar comparison of performance as above is shown in Fig.4(b). As the source data and target data vary quite a lot, SDM performs worse than the situation caused by time. LuMA still gets a good result but the best result is achieved by LSM. It is possible that different devices have an approximate linear capacities of sensing the same signals.

Now we consider the transfer learning over both time and device factors. Since this is the most common situation that we face in a WILS, the performance of LuMA is evaluated with different source data and targets. We first remain the source data collected by IBM R60 laptop at 9:30 AM, and the target is the O2 smartphone at 5:30 PM. The performance of SDM shown in Fig.5(a) is nearly 20%, due to the



(a) Laptop as source, smartphone as target.



(b) Smartphone as source, laptop as target.

Figure 5. Localization accuracy affected by both time and devices.

serious affection of both time and device. As the time factor is added, the linear relationship of different devices is broken, so LuMA performs much better than LSM. Second, we use the data collected by the O2 smartphone at 9:30 AM as source data to test the IBM laptop at 5:30 PM. The accuracy shown in Fig.5(b) is rather low, because the signals received by the laptop are quite fluctuant and make the localization hard to be accurate. However, LuMA also gets the highest accuracy when the source and target have changed.

Therefore, once we have collected a complete set of source data, the transfer learning problem can be easily solved by marking some benchmarks in the target domain. The only limit of LuMA is that the unlabeled data in the target domain should cover the whole map as much as possible, and the impact of the unlabeled data is part of our future work.

6 Conclusion and Future Work

In this paper, we propose an algorithm LuMA based on manifolds alignment to solve the transfer learning problem caused by both time and devices in Wi-Fi based indoor localization domain. LuMA utilizes source data and some target data as benchmarks to learn a corresponding relationship in a low-dimensional space, then the relationship is used to transfer knowledge from source domain to help the classification in target domain. The on-line performance evaluations show that LuMA performs better than the baselines and is adaptive in a real-world WILS. In the future, we will consider the transfer learning on spatial space, and furthermore, spatial, temporal and device space together.

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[3] J. Ham, D. Lee, and L. Saul. Semisupervised alignment of manifolds. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics 2004*, 2004.

[4] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1989.

[5] J. Letchner, D. Fox, and A. LaMarca. Large-scale localization from wireless signal strength. In *Proceedings of the 20rd AAAI Conference on Artificial Intelligence (AAAI-05)*, pages 15–20, 2005.

[6] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, 2008.

[7] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan. Adaptive localization in a dynamic wifi environment through multi-view learning. In *Proceedings of the 22rd AAAI Conference on Artificial Intelligence (AAAI-07)*, 2007.

[8] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[9] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[10] J. Yin, Q. Yang, and L. M. Ni. Learning adaptive temporal radio maps for signal-strength-based location estimation. *IEEE Transactions on Mobile Computing*, 2007.

[11] M. Youssef and A. Agrawala. The horus wlan location determination system. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 205–218, New York, NY, USA, 2005. ACM.

[12] V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan. Transferring multi-device localization models using latent multi-task learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, 2008.

[13] V. W. Zheng, E. W. Xiang, Q. Yang, and D. Shen. Transferring localization models over time. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, 2008.