Cold-Start Heterogeneous-Device Wireless Localization

Vincent W. Zheng[†], Hong Cao[‡], Shenghua Gao[♮], Aditi Adhikari[†], Miao Lin[◇], Kevin Chen-Chuan Chang[♯]

[†]Advanced Digital Sciences Center, Singapore;
[‡]McLaren Applied Technolgoies APAC, Singapore;
[♮]ShanghaiTech University, China;
[◇]Institute for Infocomm Research, A*STAR, Singapore;
[♯]University of Illinois at Urbana-Champaign, USA

Abstract

In this paper, we study a cold-start heterogeneous-device localization problem. This problem is challenging, because it results in an extreme inductive transfer learning setting, where there is only source domain data but no target domain data. This problem is also underexplored. As there is no target domain data for calibration, we aim to learn a robust feature representation only from the source domain. There is little previous work on such a robust feature learning task; besides, the existing robust feature representation proposals are both heuristic and inexpressive. As our contribution, we for the first time provide a principled and expressive robust feature representation to solve the challenging cold-start heterogeneous-device localization problem. We evaluate our model on two public real-world data sets, and show that it significantly outperforms the best baseline by 23.1%–91.3% across four pairs of heterogeneous devices.

Introduction

Indoor localization using wireless signal strength has attracted increasing interests from both research and industrial communities (Haeberlen et al. 2004; Lim et al. 2006; Zheng et al. 2008b; Xu et al. 2014). The state of the art in wireless localization is learning-based approach (Haeberlen et al. 2004; Zheng et al. 2008a). In an environment with $d_1 \in \mathbb{Z}^+$ access points (APs), a mobile device receives wireless signals from these APs. The received signal strength (RSS) values at one location are used as a feature vector $\mathbf{x} \in \mathbb{R}^{d_1}$, and the device's location is a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of possible locations in the environment. In an offline training stage, given sufficient labeled data $\{(\mathbf{x}_i, y_i)\}$, we learn a mapping function $f : \mathbb{R}^{d_1} \rightarrow \mathcal{Y}$. In an online testing stage, we use f to predict location for a new $\mathbf{x}$.

Most of the existing models work under the data homogeneity assumption, which is impractical given the prevalent *device heterogeneity*. Specifically, the assumption requires the data used in training f to have the same distribution as that used in testing. However, in practice, users carry a variety of heterogeneous mobile devices, which are different from the device used to collect data in training f. Due to different sensing chips, these heterogeneous devices easily receive different RSS values even in the same location, thus failing the model f. Denote a *surveyor device* as S, on which we collect labeled data for training the f. Denote a *target device* as T, on which we have test data to predict the labels. In Figure 1, we show that the localization accuracy can drop significantly, if we apply the model trained on S's training data to T's test data (denoted as $S \rightarrow T$), other than the same S's test data (denoted as $S \rightarrow S$).

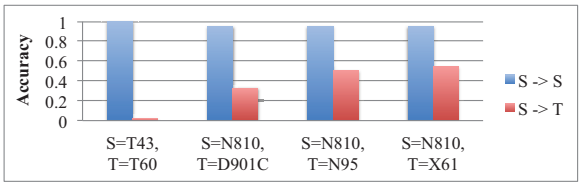


Figure 1: Accuracy drops from same-device ($S \rightarrow S$) to heterogeneous-device ($S \rightarrow T$) localization, on four pairs of devices. Data details are given in the experiment. We used SVM (Chang and Lin 2011) as the localization model.

In addition to the device heterogeneity, we face a more challenging *cold start* scenario. Some pioneers have tried to address the device heterogeneity, but they often assume that the user is willing to help collect a sufficient amount of calibration data (Haeberlen et al. 2004; Zheng et al. 2008a; Zhang et al. 2013) to tune their models before localization service is provided. Such an assumption is impractical – in fact, we often face the cold start, i.e., no calibration data is available for a new device at a new place.

In this paper, we aim to solve the following problem:

Problem 1 (Cold-start heterogeneous-device localization)**.** *Given a surveyor device S, by which we collect sufficient labeled training data $D_s = \{(\mathbf{x}_s^{(i)}, y_s^{(i)}) | i = 1, ..., n_s\}$, we aim to train a localization model f from D_s, such that f can accurately predict locations for the online test data $D_t = \{\mathbf{x}_t^{(i)} | i = 1, ..., n_t\}$ from a heterogeneous target device T. As a cold start, there is no data from T to assist training f.*

Problem 1 is challenging. Generally, we can characterize Problem 1 as an *inductive transfer learning* problem in

a *specific domain* (i.e., heterogeneous-device localization) with an *extreme setting* (i.e., cold-start). To make a better analogy with transfer learning, we interchangeably refer to surveyor device as source domain, and target device as target domain. Our extreme setting fails most of the existing transfer learning methods, which often require at least some target domain data to be available for calibration (Haeberlen et al. 2004; Zheng et al. 2008a; Tsui, Chuang, and Chu 2009; Park et al. 2011; Zhang et al. 2013).

Problem 1 is underexplored. Since there is no target domain data, the only solution to Problem 1 is to find a robust feature representation, by which we can train f with D_s and test f with D_t. To the best of our knowledge, there is little previous work for Problem 1: for example, in (Kjaergaard and Munk 2008), the authors proposed to use the RSS value ratio between every pair of APs as the robust feature representation; while in (Yedavalli et al. 2005), the authors used the RSS value ranking between every pair of APs as the robust feature representation. As we can see, the previous robust feature proposals are: 1) *heuristic*, thus lacking a formal guarantee of robustness; 2) *inexpressive*, since they are limited to using pairwise RSS comparison as features, which are only second-order w.r.t. the number of APs and not discriminative for locations (to show later).

We are looking for a *principled* and *expressive* robust feature representation to solve Problem 1. We ask two fundamental questions: **first**, what is the desired form of such a robust feature representation? For robustness, we need to make the RSS vectors collected by different devices at the same location to have the same representation. For expressiveness, we need to involve more than two APs in the representation. We thus propose a novel *High-order Pairwise (HOP) feature* representation, which is expressive by considering multiple RSS comparisons in one feature, and provably robust by leveraging the radio propagation theory (Rappaport 1999). **Second**, how do we obtain features with the desired representation? There can be infinite features satisfying the desired representation, but not all of them can represent the data well. We thus propose a novel constrained Restricted Boltzmann Machine model to enable automatic learning of the HOP features from data. To make our HOP features more discriminative, we also integrate the robust feature learning with the localization model training.

In summary, our contributions are as follows:

- We for the first time provide a principled and expressive robust feature representation, and use it to solve the challenging cold-start heterogeneous-device localization task.
- We evaluate our model on real-world data sets, and show it significantly outperforms the best baseline by 23.1%–91.3% across four pairs of heterogeneous devices.

Related Work

Problem 1 is unique as an inductive transfer learning problem with an extreme setting when there is no data for calibration from the target domain. Depending on whether a transfer learning method requires labeled or unlabeled data from the target domain in training, we categorize the existing work into three groups as follows.

The first group of transfer learning methods, including LFT (Haeberlen et al. 2004), KDEFT (Park et al. 2011) and LatentMTL (Zheng et al. 2008a), all require target domain labeled data in training. Specifically, both LFT and KDEFT learn a feature mapping function from S to T, while LatentMTL learns a common feature representation for both S and T that can do localization well.

The second group of transfer learning methods, including ULFT (Tsui, Chuang, and Chu 2009), KLIEP (Sugiyama et al. 2008), KMM (Huang et al. 2007) and SKM (Zhang et al. 2013), all require target domain unlabeled data in training. Specifically, ULFT uses expectation maximization to iteratively label the target unlabeled data and fit the feature mapping function from S to T. KLIEP, KMM and SKM all aim to account for the data distribution difference between S and T, however they adopt different approaches: KLIEP uses instance re-weighting, KMM relies on matching the sample mean in the kernel space, while SKM tries to directly align the kernel matrices of S and T.

As the above two groups of methods both require data from the target domain for training, they are not applicable to our cold-start setting. There is little work that allows transfer learning in cold-start localization. So far as we know, HLF (Kjaergaard and Munk 2008) and ECL (Yedavalli et al. 2005) are most relevant. HLF uses RSS ratio between each pair of APs as the robust feature. However, HLF's ratio feature is limited to second order (i.e., for each pair of APs); besides, such a ratio can be still different across different devices, and thus does not fundamentally solve the heterogeneity problem. On the other hand, although ECL uses a RSS-ranking based feature representation, it still turns the ranking into pairwise comparison, which is second order; besides, it lacks a formal robustness guarantee.

Finally, we note that, although there is some general transfer learning work on cold start, they are often in different domains, such as recommendation (Li, Yang, and Xue 2009), crowd-sourcing (Zhao et al. 2014), which makes their conclusions not applicable to our problem. There is also other transfer learning work that considers different cold-start setting; e.g., zero-data learning (Larochelle, Erhan, and Bengio 2008) and zero-shot learning (Gan et al. 2015; Chang et al. 2015) study that testing has different classes with training and the testing classes have no data in training. In contrast, our testing and training have the same classes.

High-Order Pairwise (HOP) Features

As there is no target domain data, we aim to learn a robust feature representation to solve the cold-start heterogeneous-device localization problem. Specifically, for robustness, we want to learn a feature representation $g : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, such that for two RSS vectors $\mathbf{x}^s$ and $\mathbf{x}^t$ collected at the same location $\tilde{y} \in \mathcal{Y}$ by device S and device T, we have:

$$g(\mathbb{E}[\mathbf{x}^s]|y^s = \tilde{y}) = g(\mathbb{E}[\mathbf{x}^t]|y^t = \tilde{y}), \quad (1)$$

where the expectation is taken over each dimension of $\mathbf{x}$ to account for the randomness in the received signal strength from each AP. Finally, given this $g(\cdot)$, we build a new $f : \mathbb{R}^{d_2} \to \mathcal{Y}$ from only S's data to predict labels for T's data.

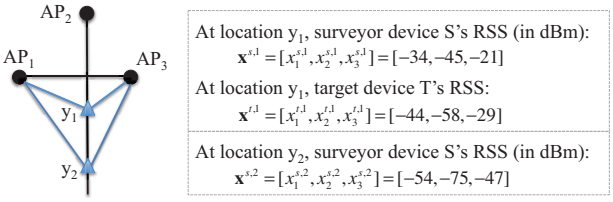


Figure 2: An example of three APs and two locations.

Insight for HOP Features

Different devices at the same location can receive different signal strengths from the same AP. Denote a device e at location y_j receives a RSS vector $\mathbf{x}^{e,j} \in \mathbb{R}^{d_1}$ from d_1 APs. For example, in Figure 2, device S at location y_1 receives a RSS vector $\mathbf{x}^{s,1} \in \mathbb{R}^3$ from three APs, and device T at y_1 receives a RSS vector $\mathbf{x}^{t,1} \in \mathbb{R}^3$. We can see that, for each AP_i, $x_i^{s,1}$ is different from $x_i^{t,1}$.

Individual RSS values' sensitivity to the device heterogeneity can be systematically explained by the radio propagation theory. Denote $z_{i|\ell}$ as a RSS value collected at a distance ℓ from AP_i. According to the *log-normal shadowing model* (Rappaport 1999), $z_{i|\ell}$ can be generated by

$$z_{i|\ell} = z_{i|\ell_0} - 10\beta \log(\frac{\ell}{\ell_0}) + \psi_i, \tag{2}$$

where $z_{i|\ell_0}$ is a RSS value from AP_i at a reference distance ℓ_0. ψ_i is a Gaussian noise with zero mean. β is a constant denoting the path loss exponent. Eq.(2) implies that a RSS value is decided by: 1) $z_{i|\ell_0}$, which is unique to device, thus explaining why different devices get different RSS values at the same location; 2) distance ℓ, such that the small ℓ is, the larger $z_{i|\ell}$ is; 3) noise ψ_i, explaining the signal fluctuation.

Pairwise RSS comparison is robust across devices. In Figure 2, we observe $x_3^{s,1} - x_1^{s,1} > 0$ and $x_3^{t,1} - x_1^{t,1} > 0$. In other words, a second-order feature representation $\delta(x_3^{e,j} - x_1^{e,j} > 0)$ is robust, where $\delta(r)$ is an indicator function returning one if r is true and zero otherwise. We can explain the robustness by Eq.(2). Intuitively, if $x_3^{s,1} - x_1^{s,1} > 0$ (i.e., signal from AP_3 is stronger), then S is generally[1] closer to AP_3 than AP_1. Hence, T at the same location is likely to see $x_3^{t,1} - x_1^{t,1} > 0$. In other words, the pairwise comparison is robust because it essentially evaluates the *relative closeness* from a location to two APs, which is device-insensitive.

Pairwise RSS comparison is not discriminative for different locations. In Figure 2, we observe $\delta(x_{k_1}^{s,1} - x_{k_2}^{s,1} > 0) = \delta(x_{k_1}^{s,2} - x_{k_2}^{s,2} > 0)$ for any AP_{k_1} and AP_{k_2} (e.g., try $k_1 = 3$, $k_2 = 1$). This means the pairwise RSS comparison feature cannot differentiate y_1 and y_2. Such a limitation is caused by the feature being only second-order and inexpressive.

Since each pairwise RSS comparison is robust, if we combine multiple pairwise RSS comparisons together, then we are still evaluating the relative closeness (which ensures the robustness), but in a higher order (which makes the feature more expressive for discriminativeness). In Figure 2, we can see $x_3^{s,1} - x_1^{s,1} > x_1^{s,1} - x_2^{s,1}$ for y_1, while $x_3^{s,2} - x_1^{s,2} < x_1^{s,2} - x_2^{s,2}$ for y_2. In other words, a third-order pairwise feature $(x_3^{e,j} - x_1^{e,j}) - (x_1^{e,j} - x_2^{e,j}) > 0$ is discriminative for y_1 and y_2. Such discriminativeness is due to our comparing: 1) a location y_j's relative closeness between AP_3 and AP_1; and 2) y_j's relative closeness between AP_1 and AP_2.

[1]We will formalize this intuition in Theorem 1.

Formulation of HOP Features

We generalize our intuition of combining multiple pairwise RSS comparisons together to formulate high-order pairwise (HOP) features. In Figure 2, the third-order pairwise feature $(x_3^{e,j} - x_1^{e,j}) - (x_1^{e,j} - x_2^{e,j}) > 0$ is a linear combination of two pairwise RSS differences. To use more APs, we consider more pairwise RSS differences. Denote a pairwise RSS difference as $(x_{k_1}^{e,j} - x_{k_2}^{e,j})$ for AP_{k_1} and AP_{k_2}. For notation simplicity, we omit the superscript ^{e,j} and only use $(x_{k_1} - x_{k_2})$ in our feature representation. Then, we linearly combine multiple pairwise RSS differences:

$$\sum\nolimits_{(k_1,k_2)} c_{k_1,k_2}(x_{k_1} - x_{k_2}) > 0, \tag{3}$$

where $c_{k_1,k_2} \in \mathbb{R}$ can be seen as the *normalized* number of times that $(x_{k_1} - x_{k_2})$ is used to construct the linear combination. The summation in Eq.(3) is over all the possible (k_1, k_2)'s, which means Eq.(3) is capable of providing the highest-order representation by using all the APs in one feature. As RSS value is noisy (c.f., the shadowing model), Eq.(3) may not always hold for different RSS samples collected over time. To deal with this random noise, we introduce $b \in \mathbb{R}$ as a "slack" to Eq.(3). Finally, we have

Definition 1 (HOP feature)**.** *A HOP feature h is defined as*

$$h = \delta\left(\sum\nolimits_{(k_1,k_2)} c_{k_1,k_2}(x_{k_1} - x_{k_2}) + b > 0\right). \tag{4}$$

HOP feature is expressive, thanks to using more APs. HOP feature is also robust, as we will prove in Theorem 1.

HOP Feature Learning and Robustness

There are many possible h's that meet Definition 1, due to the infinite choices for parameter (c_{k_1,k_2}, b)'s. However, not all of them can represent the data well. Some h can be trivial; e.g., when all the $c_{k_1,k_2} = 0$ and b is any real value, h is still robust as different $\mathbf{x}$'s are transformed to the same value. But such a h is not discriminative for locations, and thus cannot represent the data well. We propose to learn a set of h's, such that they are representative for the data:

$$\mathbf{h}^* = \underset{\mathbf{h}\in\{0,1\}^{d_2}}{\arg\max} \sum\nolimits_{k=1}^{n_L} \log P(\mathbf{x}^{(k)}; \mathbf{h}), \tag{5}$$

where $\mathbf{h} = [h_1, ..., h_{d_2}]$ is a vector of d_2 HOP features; $P(\mathbf{x}; \mathbf{h})$ is the data likelihood (to define later) with $\mathbf{h}$.

Learning by Constrained RBM

Directly learning $\mathbf{h}^*$ leads to optimizing an excessive number of parameters. Based on Eq.(4), to learn h's parameters

(c_{k_1,k_2}, b), we shall enumerate all the $(x_{k_1} - x_{k_2})$'s, which are quadratic to the number of APs. As a result, in all we have to optimize at least $O(d_1^2 \times d_2)$ parameters. Fitting more parameters generally require more data, thus increasing the labeling burden. We can reduce the number of parameters by rewriting Eq.(4) to an equivalent form, which avoids the explicit pair enumeration. As $\sum_{(k_1,k_2)} c_{k_1,k_2}(x_{k_1} - x_{k_2}) + b$ admits a linear form over different x_i's, we can organize:

$$\sum\nolimits_{(k_1,k_2)} c_{k_1,k_2}(x_{k_1} - x_{k_2}) + b = \sum\nolimits_{i=1}^{d_1} \alpha_i x_i + b, \quad (6)$$

where $\alpha_i = \sum_{(k_1,k_2)} [c_{k_1,k_2}\delta(k_1 = i) - c_{k_1,k_2}\delta(k_2 = i)]$. The weights for each x_{k_1} and x_{k_2} are c_{k_1,k_2} and $-c_{k_1,k_2}$. Therefore, by summing up weights for all the (k_1, k_2) pairs, the sum becomes zero. In other words, $\sum_{i=1}^{d_1} \alpha_i$ is zero:

$$\begin{aligned}\sum_{i=1}^{d_1} \alpha_i &\overset{1}{=} \sum_{i=1}^{d_1} \sum_{(k_1,k_2)} [c_{k_1,k_2}\delta(k_1 = i) - c_{k_1,k_2}\delta(k_2 = i)] \\ &\overset{2}{=} \sum_{(k_1,k_2)} \left[\sum_{i=1}^{d_1} c_{k_1,k_2}\delta(k_1 = i) - \sum_{i=1}^{d_1} c_{k_1,k_2}\delta(k_2 = i) \right] \\ &\overset{3}{=} \sum\nolimits_{(k_1,k_2)} (c_{k_1,k_2} - c_{k_1,k_2}) = 0,\end{aligned} \quad (7)$$

where at step 1 we plug in the definition of α_i; at step 2, we swap the summations over i and (k_1, k_2); step 3 holds, because each x_{k_1} always corresponds to one x_i. Eq.(7) implies that, a HOP feature of Eq.(4) corresponds to a special feature transformation function with constraint:

$$h = \delta\left(\sum\nolimits_{i=1}^{d_1} \alpha_i x_i + b > 0 \right), \ \ s.t. \ \sum\nolimits_{i=1}^{d_1} \alpha_i = 0. \quad (8)$$

As a result, to learn HOP features, we only need to focus on learning linear weights for each individual RSS value x_i, subject to a zero-sum constraint. In this case, we only need to optimize $O(d_1 \times d_2)$ parameters, which are several orders of magnitude smaller than the brute-force learning's $O(d_1^2 \times d_2)$ (as d_1 is often hundreds in practice).

Although there have been many feature learning methods, none of them can be directly applied to learn HOP features. According to Eq.(5) and Eq.(8), we need a generative model to learn $\mathbf{h}$, such that: 1) each $h \in \mathbf{h}$ has a binary output, based on a δ-function of a linear transformation over the numeric input x_i's; 2) the linear transformation weights α_i's have a zero-sum constraint. These two requirements for learning $\mathbf{h}$ fail the existing feature learning methods, including classic dimensionality reduction such as principle component analysis (Jolliffe 2005), kernel methods (Huang et al. 2007; Zhang et al. 2013) and deep learning such as Convolution Network (Krizhevsky, Sutskever, and Hinton 2012).

We propose a novel *constrained* Restricted Boltzmann Machine (RBM) to learn our HOP features.

First, because (Gaussian-Bernoulli) RBM (Hinton 2010) is well known as a generative model for feature learning with numeric input, binary output and linear mapping, we use RBM to instantiate the data likelihood $P(\mathbf{x}; \mathbf{h})$ in Eq.(5). In particular, RBM considers the data likelihood as

$$P(\mathbf{x}; \mathbf{h}) = \frac{1}{Z} \sum\nolimits_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}, \quad (9)$$

where $E(\mathbf{x}, \mathbf{h}) = \sum_{i=1}^{d_1} \frac{(x_i - a_i)^2}{2\pi_i^2} - \sum_{j=1}^{d_2} b_j h_j - \sum_{i,j} \frac{x_i}{\pi_i} h_j w_{ij}$ is an energy function. The first term of $E(\mathbf{x}, \mathbf{h})$ models a Gaussian distribution over each x_i, where a_i and π_i are the mean and standard deviation. The second term models the bias b_j for each h_j. The third term models the linear mapping between $\mathbf{x}$ and h_j. Finally, $Z = \sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}$ is a partition function. In RBM, each h_j can be seen as $h_j = \delta(\sum_{i=1}^{d_1} \frac{x_i}{\pi_i} w_{i,j} + b_j > 0)$, and it is sampled by a conditional probability (Krizhevsky and Hinton 2009):

$$P(h_j = 1|\mathbf{x}) = \sigma\left(\sum\nolimits_{i=1}^{d_1} \frac{x_i}{\pi_i} w_{i,j} + b_j \right), \quad (10)$$

where $\sigma(r) = \frac{1}{1+e^{-r}}$ is a sigmoid function.

Second, we extend RBM to incorporate the zero-sum constraint. We emphasize that, as this zero-sum constraint guarantees robustness (later proved in Theorem 1) and it has never been studied, our constrained RBM is a novel robust feature learning model. To take the zero-sum constraint into account, we compare Eq.(10) and Eq.(8), and set each $\alpha_i = \frac{1}{\pi_i} w_{ij}$. Denote the parameters as $\Theta = \{w_{ij}, a_i, b_j\}$. Finally, our robust feature learning minimizes:

$$L_1(\Theta) = -\frac{1}{n_L} \sum_{k=1}^{n_L} \log P(\mathbf{x}^{(k)}), \ \text{s.t.} \sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij} = 0, \forall j. \quad (11)$$

As h_j is sampled by $P(h_j = 1|\mathbf{x})$, to take this uncertainty into account, we define our robust feature representation as:

$$g(\mathbf{x}) \triangleq [P(h_1 = 1|\mathbf{x}), ..., P(h_{d_2} = 1|\mathbf{x})]. \quad (12)$$

As $g(\mathbf{x})$ is differentiable for Θ, Eq.(12) also makes it possible to integrate with localization model training for getting more discriminative HOP features, as shown later.

Robustness Guarantee

As a principled robust feature representation, we shall prove Eq.(12) meets the robustness requirement in Eq.(1).

Theorem 1 (Robustness)**.** *Given Eq.(10) subject to the constraint* $\sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij} = 0, \forall j$ *as our HOP feature, we have*

$$\forall j : P(h_j = 1|\mathbb{E}[\mathbf{x}^s]) = P(h_j = 1|\mathbb{E}[\mathbf{x}^t]).$$

Our intuition for the proof is to generalize the robustness discussion in Figure 2 from the second-order pairwise RSS comparison to the high-order features.

Proof. As $P(h_j = 1|\mathbb{E}[\mathbf{x}^s]) = \sigma\left(\sum_{i=1}^{d_1} \frac{\mathbb{E}[x_i^s]}{\pi_i} w_{i,j} + b_j \right)$ by Eq.(10), to prove $P(h_j = 1|\mathbb{E}[\mathbf{x}^s]) = P(h_j = 1|\mathbb{E}[\mathbf{x}^t])$, we only need to define $\Delta = \left(\sum_{i=1}^{d_1} \frac{\mathbb{E}[x_i^s]}{\pi_i} w_{i,j} + b_j \right) -$

$\left(\sum_{i=1}^{d_1} \frac{\mathbb{E}[x_i^t]}{\pi_i} w_{i,j} + b_j\right)$ and prove $\Delta = 0$ as below:

$$\begin{aligned}\Delta &\overset{1}{=} \sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij}(\mathbb{E}[z^s_{i|\ell_i}] - \mathbb{E}[z^t_{i|\ell_i}]) \\ &\overset{2}{=} \sum\nolimits_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij}(\mathbb{E}[z^s_{i|\ell_0} + \psi_i^{(s)}] - \mathbb{E}[z^t_{i|\ell_0} + \psi_i^{(t)}]) \\ &\overset{3}{=} \sum\nolimits_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij}[(\mu_s - 0) - (\mu_t - 0)] \\ &\overset{4}{=} (\mu_s - \mu_t) \sum\nolimits_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij} \overset{5}{=} (\mu_s - \mu_t) \cdot 0 = 0,\end{aligned}$$

where at step 1, we cancel the b_j's and let $x_i = z_{i|\ell}$ based on the shadowing model. At step 2, we plug in Eq.(2) and cancel the expectation-independent term $10\beta \log(\frac{\ell_i}{\ell_0})$. At step 3, without particular requirement on the APs, we assume $z_{i|\ell_0}$ follows a normal distribution $z_{i|\ell_0} \sim N(\mu, \sigma^2_{\ell_0})$, where μ is the mean RSS value at a reference distance ℓ_0 to different APs, $\sigma^2_{\ell_0}$ is the variance. In the shadowing model, ψ_i is a zero-mean noise. Step 5 holds as $\sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij} = 0$. □

Integration with Localization

To make our HOP features more discriminative for localization, we are inspired by (Weston, Ratle, and Collobert 2008) to consider integrating feature learning with classifier training. Specifically, as we have multiple locations, we define f as a multi-class classifier and train it with the popular *one-vs-one* setting. We let $f = \{f'_{m_1,m_2} | m_1 \in \mathcal{Y}, m_2 \in \mathcal{Y}\}$, where each $f'_{m_1,m_2} = \mathbf{v}_{m_1,m_2} \cdot g(\mathbf{x})$ is a binary classifier for locations m_1 and m_2, with $\mathbf{v}_{m_1,m_2} \in \mathbb{R}^{d_2}$ as the parameter. We emphasize how to best design f'_{m_1,m_2} is not the focus of this paper. For each f'_{m_1,m_2}, we generate a set of labels $\{y'_1, ..., y'_{n_{m_1,m_2}}\}$, where each $y'_k \in \{1, -1\}$ depends on whether an instance in D_s has its label to be m_1 or m_2. Finally, we consider a hinge loss to optimize f'_{m_1,m_2}:

$$\tilde{L}_2(f'_{m_1,m_2}) = \sum\nolimits_{k=1}^{n_{m_1,m_2}} \max\left(1 - y'_k f'_{m_1,m_2}(g(\mathbf{x}^{(k)}), 0\right).$$

Given Θ for $g(\mathbf{x})$, we define the total loss for f as:

$$L_2(f; \Theta) = \sum\nolimits_{m_1} \sum\nolimits_{m_2 : m_2 \neq m_1} \tilde{L}_2(f'_{m_1,m_2}).$$

We use voting of the f'_{m_1,m_2}'s to do prediction.

In all, we optimize the following objective function:

$$\min_{\Theta, f} \; L_1 + \lambda_2 L_2, \;\; \text{s.t.} \;\; \sum\nolimits_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij} = 0, \forall j \qquad (13)$$

where $\lambda_2 > 0$ is a parameter to tune. Recall that in Eq.(12) we defines $g(\mathbf{x})$ as differentiable to Θ, then we can optimize Eq.(13) by gradient descent on L_2. In practice, for computational convenience, we relax the constraint in Eq.(13) as a regularization term $R_1 = \frac{1}{2} \sum_{j=1}^{d_2} \left(\sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{ij}\right)^2$. Then, we optimize the regularized objective function:

$$\min_{\Theta, F} \; L_1 + \lambda_1 R_1 + \lambda_2 L_2. \qquad (14)$$

In training, we optimize Eq.(14) iteratively with Θ and F. For Θ we use contrast divergence to compute the gradient for optimization. For F we compute subgradient for the hinge loss for optimization. Due to space limit, we skip the details.

Table 1: Description for our data sets.

	#(SAMPLE)	#(LOC.)	#(AP)	#(DEVICE)	ENVIROMENT
HKUST	4,280	107	118	2	$64m \times 50m$
MIT	13,658	18	202	4	$39m \times 30m$

Table 2: Description for the devices.

	DEVICE	CATEGORY	WIRELESS CHIPSET	ROLE
HKUST	T43	IBM laptop	Intel PRO/W 2200BG	Source
	T60	IBM laptop	Intel PRO/W 3495BG	Target
MIT	N810	Nokia phone	Conexant CX3110X	Source
	X61	IBM laptop	Intel 4965AGN	Target
	D901C	Clevo laptop	Intel 5300AGN	Target
	N95	Nokia phone	TI OMAP2420	Target

Experiments

Data sets: we use two public real-world data sets: HKUST data set (Zheng et al. 2008a) and MIT data set[2] (Park et al. 2011), as shown in Tables 1 and 2. For each data set, we follow the previous work (Zheng et al. 2008a; Park et al. 2011) to choose the source device S and the target devices T's. Totally, we have four pairs of heterogeneous devices for experiments: 1) $S =$ T43, $T =$ T60, 2) $S =$ N810, $T =$ D901C, 3) $S =$ N810, $T =$ N95, 4) $S =$ N810, $T =$ X61.

In each pair of devices, for S, we randomly selected 50% of its data at each location as the labeled training data. We used the other 50% of S's data as test data ***only*** in Figure 1 to demonstrate the device heterogeneity. For T, we used 100% of its data as test data. We repeated the above process for five times, and report results with mean and standard deviation (depicted as error bars in Figures 3 and 4).

Baselines: as our cold-start problem requires no data from the target domain, in the experiment we only compare with the baselines that also fall into this setting. First, we compare with SVM (Chang and Lin 2011), which ignores device heterogeneity. Second, we compare with transfer learning methods that require no data from the target device, including HLF (Kjaergaard and Munk 2008) and ECL (Yedavalli et al. 2005), as discussed in the related work.

Evaluation metric: following the convention of localization study (Haeberlen et al. 2004; Lim et al. 2006; Xu et al. 2014), we evaluated the *accuracy under an error distance* (i.e., the accuracy when the prediction is within a certain distance to its ground truth). We will study the impact of error distance later in Figure 4(a). Unless specified, we set the error distance as 4 meters in evaluation.

Impact of Model Parameters

We study the impact of λ_1, λ_2 and d_2. In Figure 3, we fix $\lambda_2 = 1$, $d_2 = 100$ and tune λ_1. Our model tends to achieve higher accuracies when λ_1 is bigger. This means we prefer the zero-sum constraint to hold. Then, we fix $\lambda_1 = 10$, $d_2 = 100$ and tune λ_2. Our model is generally insensitive to λ_2. When $\lambda_2 = 100$, the accuracies tend to drop. This may be because a too big λ_2 makes the loss of f overwhelm the

[2]We do not use the devices N810(2) and EEE900A due to their low heterogeneity to N810 as reported in (Park et al. 2011).

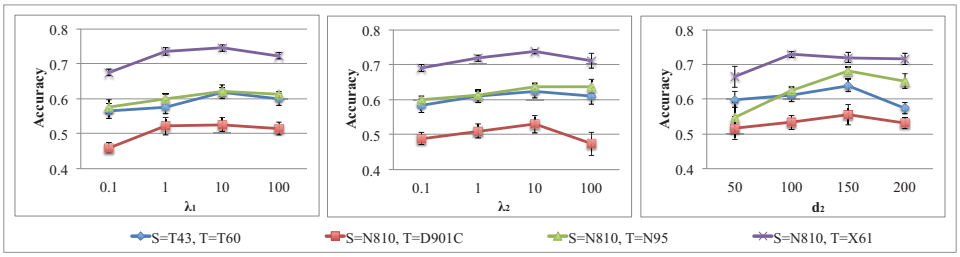
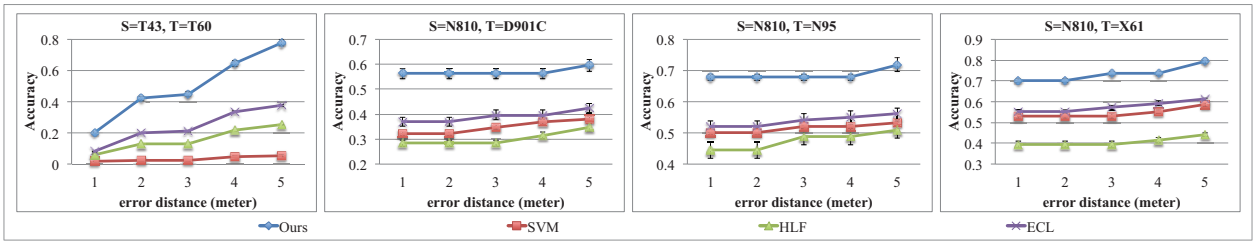
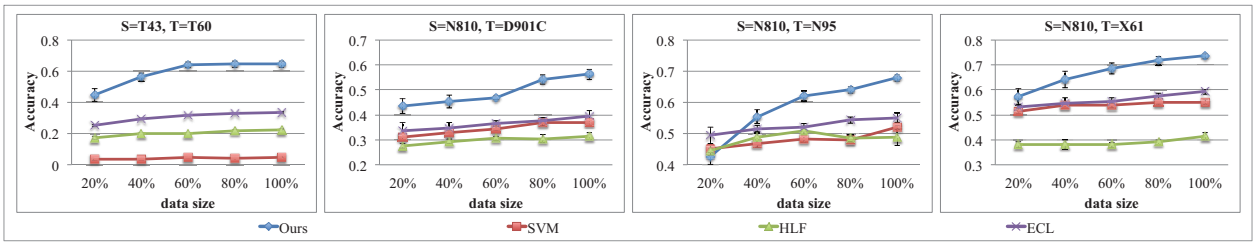


Figure 3: Impact of model parameters.



(a) Impact of error distance.



(b) Impact of training data size.

Figure 4: Comparison with baselines.

objective function. Finally, we fix $\lambda_1 = 10$, $\lambda_2 = 1$ and tune d_2. Our model tends to achieve the best accuracies when $d_2 = 150$. In practice, like other dimensionality reduction methods (Jolliffe 2005; Krizhevsky, Sutskever, and Hinton 2012), we suggest tuning d_2 empirically. In the following, we fix $\lambda_1 = 10$, $\lambda_2 = 1$ and $d_2 = 150$.

Comparison with Baselines

We first compare with the baselines by using all the training data of S, and evaluate the localization accuracy under different error distances. As shown in Figure 4(a), our model is consistently better than the baselines. Besides, we compare with the baselines by using different amount of training data from S, and evaluate the localization accuracy under 4-meter error distance. As shown in Figure 4(b), our model is in general consistently better than the baselines. In summary, by using all the training data of S and under the error distance of 4 meters, our model can achieve 23.1%–91.3% relative accuracy improvement than the best baseline (i.e, ECL) across different pairs of devices. Such improvements are all statistically significant – according to our t-tests, both one-tailed test $p_1 < 0.01$ and two-tailed test $p_2 < 0.01$.

Our model is better than SVM, as we address device heterogeneity. Our model is better than HLF, as HLF only uses pairwise RSS ratio features, which are not discriminative (as we discussed in Figure 2) and are sensitive to the RSS values. In fact, we observe that HLF is sometimes even worse than SVM. Our model is also better than ECL, since ECL is limited in using pairwise comparison (in contrast with our using higher-order pairwise features). Finally, we emphasize both HLF and ECL lack robustness guarantee as we do.

Conclusion

We studied the cold-start heterogeneous-device localization problem, where we have to train a localization model only from a surveyor device's data and test it with a heterogeneous target device's data. This problem corresponds to an extreme inductive transfer learning setting when there is no target domain data to assist training. Due to cold start, we aim to find a robust feature representation. We proposed a novel HOP feature representation, which is principled with robustness guarantee and expressive to be able to represent the data and discriminate locations. We also proposed a novel constrained RBM model to enable automatic learning of the HOP features from data. Finally, we integrated the robust feature learning with the localization model training, so as to get more discriminative HOP features and complete the localization framework. Our model can achieve 23.1%–

91.3% relative accuracy improvement than the best state-of-the-art baseline. In the future, we wish to extend our method to incorporate unlabeled data from the source domain.

Acknowledgment

This work is supported by the research grant for the Human-centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapores Agency for Science, Technology and Research (A*STAR). This work is also supported by the Shanghai Pujiang Program (No.15PJ1405700) and NSFC (No. 61502304).

References

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2(3):27:1–27:27.

Chang, X.; Yang, Y.; Hauptmann, A. G.; Xing, E. P.; and Yu, Y. 2015. Semantic concept discovery for large-scale zero-shot event detection. In *Proc, of the 24th International Joint Conference on Artificial Intelligence*, IJCAI '15, 2234–2240.

Gan, C.; Lin, M.; Yang, Y.; Zhuang, Y.; and Hauptmann, A. G. 2015. Exploring semantic inter-class relationships (SIR) for zero-shot action recognition. In *Proc. of the 29th AAAI Conference on Artificial Intelligence*, AAAI '15, 3769–3775.

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 *Proc. of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, 70–84.

Hinton, G. 2010. A practical guide to training restricted boltzmann machines. Technical report, University of Toronto.

Huang, J.; Smola, A.; Gretton, A.; Borgwardt, K. M.; and Schölkopf, B. 2007. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 20*, NIPS '07, 601–608.

Jolliffe, I. 2005. *Principal component analysis*. Wiley Online Library.

Kjaergaard, M. B., and Munk, C. V. 2008. Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength. In *Proc. of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, PerCom '08, 110–116.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, NIPS '12, 1097–1105.

Larochelle, H.; Erhan, D.; and Bengio, Y. 2008. Zero-data learning of new tasks. In *Proc. of the 23rd National Conference on Artificial Intelligence*, AAAI '08, 646–651.

Li, B.; Yang, Q.; and Xue, X. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proc. of the 26th Annual International Conference on Machine Learning*, ICML '09, 617–624.

Lim, H.; Kung, L.-C.; Hou, J. C.; and Luo, H. 2006. Zero-configuration, robust indoor localization: Theory and experimentation. In *Proc. of the 25th IEEE International Conference on Computer Communications*, INFOCOM '06, 1–12.

Park, J.-G.; Curtis, D.; Teller, S. J.; and Ledlie, J. 2011. Implications of device diversity for organic localization. In *Proc. of the 30th IEEE International Conference on Computer Communications*, INFOCOM '11, 3182–3190.

Rappaport, T. S. 1999. *Wireless communications: principles and practice*. Prentice Hall.

Sugiyama, M.; Nakajima, S.; Kashima, H.; Bünau, P.; and Kawanabe, M. 2008. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 21*, NIPS '08, 1433–1440.

Tsui, A. W.; Chuang, Y.-H.; and Chu, H.-H. 2009. Unsupervised learning for solving rss hardware variance problem in wifi localization. *Mobile Networks and Applications* 14(5):677–691.

Weston, J.; Ratle, F.; and Collobert, R. 2008. Deep learning via semi-supervised embedding. In *Proc. of the 25th international conference on Machine learning*, ICML '08, 1168–1175.

Xu, N.; Low, K. H.; Chen, J.; Lim, K. K.; and Ozgul, E. B. 2014. Gp-localize: Persistent mobile robot localization using online sparse gaussian process observation model. In *Proc. of the 28th AAAI Conference on Artificial Intelligence*, AAAI '14, 2585–2593.

Yedavalli, K.; Krishnamachari, B.; Ravula, S.; and Srinivasan, B. 2005. Ecolocation: A sequence based technique for rf localization in wireless sensor networks. In *Proc. of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, 285–292.

Zhang, K.; Zheng, V. W.; Wang, Q.; Kwok, J.; Yang, Q.; and Marsic, I. 2013. Covariate shift in hilbert space: A solution via sorrogate kernels. In *Proc. of the 30th international conference on Machine learning*, ICML '13, 388–395.

Zhao, Z.; Cheng, J.; Wei, F.; Zhou, M.; Ng, W.; and Wu, Y. 2014. Socialtransfer: Transferring social knowledge for cold-start cowdsourcing. In *Proc. of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, 779–788.

Zheng, V. W.; Pan, S. J.; Yang, Q.; and Pan, J. J. 2008a. Transferring multi-device localization models using latent multi-task learning. In *Proc. of the 23rd AAAI Conference on Artificial Intelligence*, AAAI '08, 1427–1432.

Zheng, V. W.; Xiang, E. W.; Yang, Q.; and Shen, D. 2008b. Transferring localization models over time. In *Proc. of the 23rd AAAI Conference on Artificial Intelligence*, AAAI '08, 1421–1426.