Cold-Start Heterogeneous-Device Wireless Localization

Vincent W. Zheng, Hong Cao, Shenghua Gao, Aditi Adhikari, Miao Lin, Kevin Chen-Chuan Chang

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 $d_1$ 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 $x \in \mathbb{R}^{d_1}$, 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 $\{(x_i, y_i)\}$, 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 $x$.

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 $S \rightarrow S$).

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 $D_s = \{(x_s^{(i)}, y_s^{(i)}) | i = 1, ..., n_s\}$, we aim to train a localization model $f$ from $D_s$, such that $f$ can accurately predict locations for the online test data $D_t = \{x_t^{(i)} | i = 1, ..., n_t\}$ 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), KLIIEP (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 $T$. KLIIEP, KMM and SKM all aim to account for the data distribution difference between $S$ and $T$, however they adopt different approaches: KLIIEP 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 $g : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}$, such that for two RSS vectors $x^s$ and $x^t$ collected at the same location $\tilde{y} \in \mathcal{Y}$ by device $S$ and device $T$, we have:

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

where the expectation is taken over each dimension of $x$ to account for the randomness in the received signal strength from each AP. Finally, given this $g(\cdot)$, we build a new $f : \mathbb{R}^{d_2} \rightarrow \mathcal{Y}$ from only $S$’s data to predict labels for $T$’s data.
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 $y_e$ receives a RSS vector $x^{e-j}$ in $\mathbb{R}^d$ from $d_1$ APs. For example, in Figure 2, device $S$ at location $y_1$ receives a RSS vector $x^{s-1} \in \mathbb{R}^3$ from three APs, and device $T$ at $y_1$ receives a RSS vector $x^{t-1} \in \mathbb{R}^3$. We can see that, for each $AP_i$, $x^{i-1} \in \mathbb{R}^3$ is different from $x^{j-1} \in \mathbb{R}^3$.

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

$$z_{i|e} = z_{i|\ell_0} - \beta \log(\ell_0) + \psi_i,$$

where $z_{i|\ell_0}$ is a RSS value from $AP_i$ at a reference distance $\ell_0$, $\psi_i$ is a Gaussian noise with zero mean. $\beta$ 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 $\ell$, such that the small $\ell$ is, the larger $z_{i|\ell}$ is; 3) noise $\psi_i$, explaining the signal fluctuation.

Pairwise RSS comparison is robust across devices. In Figure 2, we observe $x^{s-1}_3 - x^{j-1}_1 > 0$ and $x^{t-1}_3 - x^{j-1}_1 > 0$. In other words, a second-order feature representation $\delta(x^{s,j}_3 - x^{e,j}_1 > 0)$ 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 $x^{s-1}_3 - x^{j-1}_1 > 0$ (i.e., signal from $AP_3$ is stronger), then $S$ is generally closer to $AP_3$ than $AP_1$. Hence, $T$ at the same location is likely to see $x^{t-1}_3 - x^{j-1}_1 > 0$. 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 $\delta(x^{s-1}_3 - x^{j-1}_1 > 0) = \delta(x^{s,2}_1 - x^{f,2}_2 > 0)$ for any $AP_1$, and $AP_2$ (e.g., try $k_1 = 3$, $k_2 = 3$). 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 $x^{s,1}_3 - x^{s,1}_1 > x^{t,1}_1 - x^{s,1}_2$ for $y_1$, while $x^{s,2}_3 - x^{s,2}_1 < x^{t,2}_1 - x^{s,2}_2$ for $y_2$. In other words, a third-order pairwise feature $\delta(x^{s,2}_3 - x^{e,2}_3) - (x^{s,2}_1 - x^{e,2}_2) > 0$ is discriminative for $y_1$ and $y_2$. Such discriminativeness is due to our comparing: 1) a location $y_j$’s relative closeness between $AP_1$ and $AP_2$; and 2) $y_j$’s relative closeness between $AP_1$ and $AP_2$.

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 $\delta(x^{s,j}_3 - x^{e,j}_1) - (x^{s,j}_1 - x^{e,j}_2) > 0$ 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 $(x^{c,j}_k - x^{c,j}_b)$ for $AP_k$ and $AP_b$. For notation simplicity, we omit the superscript $c,j$ and only use $(x_k - x_b)$ in our feature representation. Then, we linearly combine multiple pairwise RSS differences:

$$\sum_{(k_1,k_2)} c_{k_1,k_2} (x_{k_1} - x_{k_2}) > 0,$$

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

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

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 $x$’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:

$$h^* = \arg\max_{h \in \{0,1\}^{d_2}} \sum_{k=1}^{n_h} \log P(x^{(k)}; h),$$

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

Learning by Constrained RBM

Directly learning $h^*$ leads to optimizing an excessive number of parameters. Based on Eq.(4), to learn $h$’s parameters
where $E(\mathbf{x}, \mathbf{h}) = \sum_{i=1}^{d_1} \frac{(x_i - \mu_i)^2}{2\sigma_i^2} - \sum_{j=1}^{d_2} b_j h_j - \sum_{i,j} \frac{1}{\pi_i} w_{i,j} h_j$ is an energy function. The first term of $E(\mathbf{x}, \mathbf{h})$ models a Gaussian distribution over each $x_i$, where $\mu_i$ and $\sigma_i$ 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 $h_j = \delta(\sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{i,j} + b_j > 0)$, and it is sampled by a conditional probability (Krizhevsky and Hinton 2009):

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

(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 $\alpha_i = \frac{1}{\pi_i} w_{i,j}$. Denote the parameters as $\Theta = \{w_{i,j}, \alpha_i, b_j\}$.

Finally, our robust feature learning minimizes:

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

(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:

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

(12)

As $g(\mathbf{x})$ is differentiable for $\Theta$, 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 $\sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{i,j} = 0, \forall j$ as our HOP feature, we have

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

(13)

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 $P(h_j = 1|\mathbb{E}[\mathbf{x}]) = \sigma \left( \sum_{i=1}^{d_1} \frac{\mathbb{E}[x_i]}{\pi_i} w_{i,j} + b_j \right)$ by Eq.(10), to prove $P(h_j = 1|\mathbb{E}[\mathbf{x}]) = P(h_j = 1|\mathbb{E}[\mathbf{x}])$, we only need to define $\Delta = \left( \sum_{i=1}^{d_1} \frac{\mathbb{E}[x_i]}{\pi_i} w_{i,j} + b_j \right) -$
\[(\sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{i,j} + b_j) \text{ and prove } \Delta = 0 \text{ as below:}
\]
\[
\Delta = \frac{1}{\pi_i} w_{i,j} (E[z_{i,\ell_0}^t] - E[z_{i,\ell_0}^t]) + \frac{1}{\pi_i} w_{i,j} (E[z_{i,\ell_0}^t] + \psi_i^t) - E[z_{i,\ell_0}^t] + \psi_i^t) \\
\Rightarrow \frac{1}{\pi_i} w_{i,j} (E[z_{i,\ell_0}^t] - E[z_{i,\ell_0}^t]) + \frac{1}{\pi_i} w_{i,j} (E[z_{i,\ell_0}^t] + \psi_i^t) - E[z_{i,\ell_0}^t] + \psi_i^t) \\
\Rightarrow \frac{1}{\pi_i} w_{i,j} (\mu_s - \mu_t) = 0, \]

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

**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 \(f = \{f'_{m_1, m_2} \in \mathcal{Y}| m_1 \in \mathcal{Y}, m_2 \in \mathcal{Y}\}\), where each \(f'_{m_1, m_2} = \mathbf{v}_{m_1, m_2} \cdot g(x)\) 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 \(f'_{m_1, m_2}\):

\[
\hat{L}_2(f'_{m_1, m_2}) = \sum_{k=1}^{n_{m_1, m_2}} \max\left(1 - y_k f'_{m_1, m_2}(g(x^{(k)})), 0\right)
\]

Given \(\Theta\) for \(g(x)\), we define the total loss for \(f\) as:

\[
L_2(f; \Theta) = \sum_{m_1} \sum_{m_2 \neq m_1} \hat{L}_2(f'_{m_1, m_2}).
\]

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

In all, we optimize the following objective function:

\[
\min_{\Theta, f} L_1 + \lambda_2 L_2, \text{ s.t. } \sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{i,j} = 0, \forall j \quad (13)
\]

where \(\lambda_2 > 0\) is a parameter to tune. Recall that in Eq.(12) we define \(g(x)\) as differentiable to \(\Theta\), 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 \(R_1 = \frac{1}{2} \sum_{j=1}^{d_2} \left(\sum_{i=1}^{d_1} \frac{1}{\pi_i} w_{i,j}\right)^2\). Then, we optimize the regularized objective function:

\[
\min_{\Theta, F} L_1 + \lambda_1 R_1 + \lambda_2 L_2. \quad (14)
\]

In training, we optimize Eq.(14) iteratively with \(\Theta\) and \(F\). For \(\Theta\) 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.**

<table>
<thead>
<tr>
<th>#SAMPLE</th>
<th>#LOC.</th>
<th>#AP</th>
<th>#DEVICE</th>
<th>ENVIRONMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKUST</td>
<td>4,280</td>
<td>107</td>
<td>2</td>
<td>64m × 50m</td>
</tr>
<tr>
<td>MIT</td>
<td>13,658</td>
<td>18</td>
<td>202</td>
<td>39m × 30m</td>
</tr>
</tbody>
</table>

**Table 2: Description for the devices.**

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

**Experiments**

**Data sets:** we use two public real-world data sets: HKUST data set (Zhang et al. 2008a) and MIT data set² (Park et al. 2011), as shown in Tables 1 and 2. For each data set, we follow the previous work (Zhang 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 heterogenous 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 (Haebeler 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 \(\lambda_1, \lambda_2\) and \(d_2\). In Figure 3, we fix \(\lambda_2 = 1, d_2 = 100\) and tune \(\lambda_1\). Our model tends to achieve higher accuracies when \(\lambda_1\) is bigger. This means we prefer the zero-sum constraint to hold. Then, we fix \(\lambda_1 = 10, d_2 = 100\) and tune \(\lambda_2\). Our model is generally insensitive to \(\lambda_2\). When \(\lambda_2 = 100\), the accuracies tend to drop. This may be because a too big \(\lambda_2\) makes the loss of \(f\) overwhelm the

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


