Learning to Localize

[Draft - v3.0 - Jun 9, 2011 - changes made in the 'Evaluation' section]

AG
Computer Science
Stony Brook University
agoswami@cs.sunysb.edu

SD
Computer Science
Stony Brook University
samir@cs.sunysb.edu

LO
Computer Science
Stony Brook University
leortiz@cs.sunysb.edu

ABSTRACT

We consider the problem of localizing a wireless client in an indoor environment based on the signal strength of its transmitted packets as received on stationary sniffers or access points.

Current state-of-the art indoor localization techniques have the drawback that they rely extensively on a 'training phase'. This 'training' is a labor intensive process and must be done for each target-area under consideration for various device types. This clearly does not scale for large target areas. The introduction of unmodeled hardware with heterogeneous power-levels etc further reduces the accuracy of these techniques.

We propose a solution in which we model the received signal strength as a Gaussian Mixture Model (GMM). We use expectation maximization to find the parameters of our GMM. We can now give a location fix for a transmitting device based on the maximum likelihood estimate. This way, we not only avoid the costly 'training phase' but also make our location estimates much more robust in the face of various form of heterogeneity and time varying phenomena. We present our results on two different indoor testbeds (CEWIT and Computer Science Buildings in Stony Brook University) with multiple WiFi devices (iphones, android, laptops, netbooks). We demonstrate that the accuracy is at par with state-of-the-art techniques but without requiring any training.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Delphi theory

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

The increasing use of wireless networking has fueled the need for location-aware pervasive computing applications in indoor environments. Traditional GPS-based techniques have problems working indoor which make them unattractive for such fine-grained indoor localization. On the other hand, indoor wireless LAN (WLAN) technologies, which have been enthusiastically and widely adopted in enterprises and homes, give us interesting features like Received Signal Strength(RSS), Angle of Arrival(AoA) etc for robust location estimation. Received signal strength (RSS) is particularly interesting because current commercial hardware can be used to extract the signal strength of wireless frames being transmitted by a Wi-Fi device.

Several techniques [x, y, x] have demonstrated the viability of using the RSS metric for location estimation. It is interesting to note here that most of these location-estimation systems can essentially be categorized in two distinct ways : a client-based approach [p, q, r] and an infrastructure-based approach [a, b, c]. In the client-based approach, the client device measures the signal strength as seen by it from various AP(Access Point). This information is used to locate the client. In the infrastructure-based approach, the network administrator can use simple sniffing devices (or APs masquerading as sniffers) to monitor clients and extract the RSS from the tx-client. This sniffed information is used to locate the client. Considering ease of management, provisioning, security, deployment, maintenance etc, the infrastructure-based model seems alluring for large-scale deployments, especially if building and maintaining the model can be automated. Moreover, such techniques perform location estimation without requiring hardware and/or software changes on the client device, which make them particularly attractive.

Most of the existing solutions for WLAN location estimation work in two phases. The first phase is a pre-deployment *'offline phase'* aimed at building detailed RF maps or RF propagation models based on a survey of the target location. The second phase is an *'online phase'* of location estimation, where a localization algorithm is used to give a location estimate for an observed set of signal strength measurements. All such techniques suffer from three major drawbacks that serve as the motivation for this present work :

First, the WiFi hardware variance problem : the device used during the *'offline phase'* may differ from the target device in the *'online phase'*. Unmodeled hardware devices operating at different power-levels can introduce significant variations in the signal patterns between the training device and the target device. This adversely affects the accuracy of location estimation. Experiments described later in this paper indicate how hardware variance between four common commodity WiFi devices can significantly degrade the positional accuracy of two commonly used localization algorithms.

Second, the *'offline phase'* requires an extensive pre-deployment effort which usually involves labor-intensive sampling of signal strength values at discretized locations in the target space. Again, through experiments we show that location accuracy depends significantly on the granularity of the training positions. If the training positions are coarse grained, the location estimates become substantially poorer.

Third, static models built during the *'offline phase'* can substantially reduce the accuracy of location estimates in the presence of time varying phenomena like movement of people inside the building, changing environmental and occupancy conditions etc. The heavy pre-deployment effort makes such models difficult to maintain and update.

In this work, we propose a novel indoor localization algorithm, GEM, that tries to leverage the infrastructure based model of location determination systems while eliminating any pre-deployment effort. Packet transmissions made by a client are received on stationary sniffers or access points which extract the signal strength and mac-id of the client, and report this information to a central localization server. Using this information, GEM builds a model for that device and gives a location estimate based on the model.

GEM provides several key benefits by eliminating the *'offline phase'*. First, by building a model for each target device effectively addresses the hardware variance problem. Thus GEM can be used across heterogeneous devices, each operating at different power levels. Second, no pre-deployment effort makes GEM particularly attractive for large target spaces like malls, office spaces etc Third, GEM is an online algorithm : the model parameters get updated and modified based on real-time signal strength observations. . As such, GEM is able to adapt to dynamic changes in the target space.

Our results of deploying GEM in two different office buildings are promising. We specifically note that when measurements made using one device are used to localize a different device, GEM is seen to perform better that RF signal map based techniques like RADAR[x] and Probabilistic[y]

2. RELATED WORK

Some calibration-free techniques have been proposed [7] [11] [15] etc. The objective of such techniques is to automate the effect of wireless physical characteristics on RSS measurements and make them responsive to environmental dynamics like temperature and humidity variations, furniture variation, human mobility etc. This is usually done by having reference Access Points (or sniffers) deployed in the target space and then measuring RSS between the 802.11 APs and also between a client and its neighbouring APs (or sniffers). In [7] Moares et al use an indoor signal propagation model to generate a *radio propagation map (RPM)* at each sniffer. Thereafter they use RSS measurements between the sniffers and a reference Access Point(AP) to reconstruct the RPM, either periodically or when there are significant variations of RSS values. In [15] Lim et al. use the on-line RSS measurements to create a mapping between the RSS measure and the actual geographical distance.

Such techniques are essentially modelled to capture real-time changes in the environmental dynamics of the target space. But they do not model variations in client hardware and transmission power which can significantly degrade the positional accuracy of RSS based Wi-Fi localization schemes.

In [20] Tsui et al. also observe that hardware variance can significantly degrade the positional accuracy of RSS-based Wi-Fi localization systems. Infact they note that the hardware variance problem is not limited to differences in the WiFi chipsets used by training and tracking devices but also occurs when the same Wi-Fi chipsets are connected to different antenna types and/or packaged in different encapsulation materials. The authors stick to the *online*-training and *offline* location-determination model but add an intermediate online-adjustment phase . In this intermediate phase they use unsupervised learning methods to construct a signal transformation function between the training device and a new tracked device.

In [19] Tao et al. have an interesting take on unmodelled-hardware and transmission power variations being effected by a transmitting client. They also stick to the *online*-training and *offline* location-determination model. However, they observe that RSS is linearly proportional to transmission power. Thus the difference in received signal strengths between a pair of sniffer devices would not vary dramatically as the transmission power of a client device changes. Based on the difference in signal strength between every pair of sniffers, they suggest a weighted heuristic to estimate a location-fix for a given target RSS fingerprint. With such a 'difference' based approach, we can no longer assume that the sniffers are independent. Thus, we are restricted to the use of a heuristic in this model. However, the observation that RSS is linearly proportional to transmission power is very interesting. Infact, we use this observation in building our model.

The major contribution of this work is to develop an algorithm that does not rely on training data. Instead, the algorithm can learn the parameters of the model from real-time transmissions being made by a Tx-client. Thus it can adapt to variations in transmit power across heterogeneous devices which makes it particularly suitable for server-side localization techniques. Plus this model can also factor in real-time changes in the environmental dynamics of the target space.

3. WIRELESS CHARACTERISTICS

Our system is based on the 802.11 wireless networking protocol, which is inexpensive and widely deployed in enterprise offices and academic campuses. 802.11 uses 11 channels in the ISM band. Signal propagation in this band is complex and in this section, we identify the different causes of variation in the wireless channel quality and how we factor them into our model. Our approach is server-based, where we capture client packets using sniffers. As such, we are mainly concerned with the variations that affect the Received Signal Strength (RSS) on the sniffer. In this section, experimentally validate two observations that have been made previously in wireless-localization literature. We model our problem around these two observations.

3.1 Distribution of Signal Strength

Figure 1 shows the distribution of Received Signal Strength values observed by a sniffer located a fixed distance apart from a transmitting client. The Tx-client is a Dell laptop having a Ubiquiti XR2 wireless card and is using a fixed power-level for wireless transmissions.

We observe that the Signal Strength distribution is roughly Gaussian. In [19] et al also make similar observations. [12] [7] etc also model signal intensity as a normal distribution.

3.2 Transmission Power

Figure 2 shows how the observed signal strength changes as the transmission power is varied. Our experiments validate the observations made in [19] by Tao et al in that the observed signal strength is linearly proportional to the transmission power.

4. PROBLEM FORMULATION

The Gaussian Mixture Model is a simple linear superposition of Gaussian components, aimed at providing a richer class of density models than the single Gaussian. We now formulate our problem as a Gaussian mixture in terms of discrete latent variables.

4.1 Latent Variables for Target Locations and Power Levels

We introduce a J-dimensional binary random variable $\mathbf{x}$ representing possible target locations. $\mathbf{x}$ has a 1-of-J representation in which a particular element x_j is equal to one and all other elements are equal to 0. The values of x_j therefore satisfy $x_j \in \{0,1\}$ and $\sum_j x_j = 1$. Thus we see that there are J possible states for the vector $\mathbf{x}$

The probability distribution over $\mathbf{x}$ can be specified as a multinomial

$$p(x_j = 1) = \upsilon_j \tag{1}$$

where the parameters $\{\upsilon_j\}$ must satisfy

$$0 \leq \upsilon_j \leq 1 \; and \; \sum_{j=0}^{J} \upsilon_j = 1 \tag{2}$$

Similarly, let us introduce a K-dimensional binary random variable $\mathbf{z}$ representing Power Levels. $\mathbf{z}$ has a 1-of-K representation in which a particular element z_k is equal to one and all other elements are equal to 0. The values of z_k therefore satisfy $z_k \in \{0,1\}$ and $\sum_k z_k = 1$. Vector $\mathbf{z}$ has K possible states.

The distribution over $\mathbf{z}$ is specified as a multinomial

$$p(z_k = 1) = \tau_k \tag{3}$$

where the parameters $\{\tau_k\}$ must satisfy

$$0 \leq \tau_k \leq 1 \; and \; \sum_{k=0}^{K} \tau_k = 1 \tag{4}$$

4.2 Constructing the distribution over the observed signal strengths

Let $\mathbf{s}$ be the N-dimensional vector representing the signal strengths observed by the N sniffers placed in the area.

Using the chain rule of probability, we can now define the joint distribution $p(\mathbf{s}, \mathbf{x}, \mathbf{z})$ in terms of the distribution $p(\mathbf{x}, \mathbf{z})$ and the conditional distribution $p(\mathbf{s}|\mathbf{x}, \mathbf{z})$, corresponding to the graphical model in Figure 3.

$$p(\mathbf{s}, \mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z}) p(\mathbf{s}|\mathbf{x}, \mathbf{z}) \tag{5}$$

Moreover $\mathbf{x}$ and $\mathbf{z}$ are independent random variables. So we have

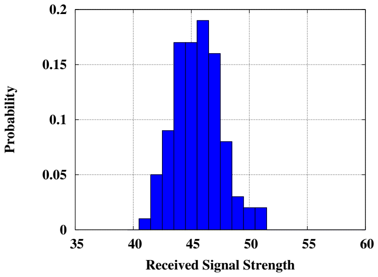


Figure 1: The distribution of RSS observed on a sniffer

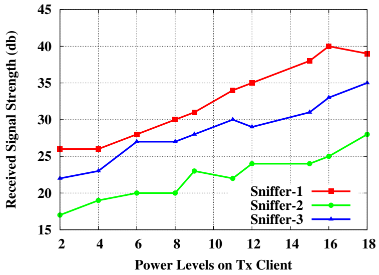


Figure 2: RSS as a function of the Tx-power of a device.

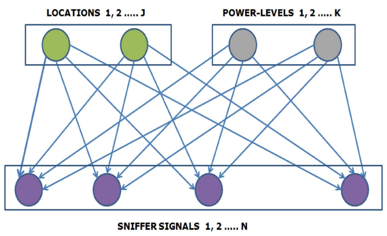


Figure 3: The GMM for our problem

$$\begin{aligned} p(\mathbf{s},\mathbf{x},\mathbf{z}) &= p(\mathbf{x},\mathbf{z})p(\mathbf{s}|\mathbf{x},\mathbf{z}) \\ &= p(\mathbf{x})p(\mathbf{z})p(\mathbf{s}|\mathbf{x},\mathbf{z}) \end{aligned} \quad (6)$$

Equation 6 gives us the joint distribution of $p(\mathbf{s},\mathbf{x},\mathbf{z})$. The marginal distribution of $\mathbf{s}$ is then obtained by summing the joint distribution over all possible states of $\mathbf{x}$ and $\mathbf{z}$ to give the following probabilistic model :

$$p(\mathbf{s}) = \sum_{\mathbf{x}}\sum_{\mathbf{z}} p(\mathbf{x})p(\mathbf{z})p(\mathbf{s}|\mathbf{x},\mathbf{z}) \quad (7)$$

4.2.1 *Independence of Sniffers*

We assume the sniffers are independent. This assumption is justified in our model because our sniffers are passive nodes responsible for capturing wireless packets. They have no interaction with each other.

Thus, the term $p(\mathbf{s}|\mathbf{x},\mathbf{z})$ in equation 7 can be simplified as

$$p(\mathbf{s}|\mathbf{x},\mathbf{z}) = \prod_{i=1}^{N} p(s_i|\mathbf{x},\mathbf{z}) \quad (8)$$

Moreover, from the observations made about Signal Strength variations in Section 3.1 above, the distribution of signal strength can be modelled as a Gaussian determined by the (location, power-level) pair.

That is

$$s_i|(x_j, z_k) \sim gaussian(\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}) \quad (9)$$

This lends simplicity to our model since the term $p(\mathbf{s}|\mathbf{x},\mathbf{z})$ in equation 8 can be further simplified as

$$p(\mathbf{s}|\mathbf{x},\mathbf{z}) = \sum_{j=1}^{J}\sum_{k=1}^{K}(\prod_{i=1}^{N}\mathcal{N}[s_i|\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}]) \quad (10)$$

4.3 Model Parameters

Putting equation 7 and equation 10 together we get the distribution of $\mathbf{s}$ as

$$p(\mathbf{s}) = \sum_{j=1}^{J}\sum_{k=1}^{K}(\upsilon_j\tau_k\prod_{i=1}^{N}\mathcal{N}[s_i|\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}]) \quad (11)$$

Thus we have modelled the marginal distribution of $\mathbf{s}$ as a Gaussian mixture with target locations and power levels as our latent variables. The parameters of our model are

$$\theta = \left(\upsilon_j, \tau_k, (\mu_{i\ (j,k)}, \sigma_{i\ (j,k)})\right) \quad (12)$$

where $j \in \{1,...J\}$, $k \in \{1,...K\}$ and $i \in \{1,...N\}$. We now use the Expectation Maximization(EM) algorithm to estimate the parameters of our model.

5. EM ALGORITHM

An elegant and powerful method for finding maximum likelihood solutions for models with latent variables is the Expectation Maximization(EM) algorithm. The EM algorithm is an iterative process through two steps: an expectation step(E-step) and a maximization step(M-step). During the iterations, a sequence of model parameters $\theta^{\mathbf{0}}$, $\theta^{\mathbf{1}}$, ..., θ^{*} is generated where $\theta^{\mathbf{0}}$ is the initial parameter and θ^{*} is the converged parameter obtained when the algorithm terminates.

5.1 E-step

Suppose we have a data set of observations $\overline{\mathbf{S}} = \{\ \mathbf{s}^0$, $\mathbf{s}^1$, ...,$\mathbf{s}^M\}$. The E-step corresponds to finding the expected value of the hidden component ($\mathbf{x}$ and $\mathbf{z}$) values given the observed data $\overline{\mathbf{S}}$ and the current parameter estimates.

Using this observation set and the current parameter estimates, we find out the posterior probabilities (or responsibilities) as follows.

For each observation $\mathbf{s}^l$

$$\pi^{l}_{(x_j,z_k)} = p(x_j = 1, z_k = 1|\mathbf{s}^l) \quad (13)$$

$$= \frac{p(x_j = 1)p(z_k = 1)p(\mathbf{s}^l|x_j = 1, z_k = 1)}{\sum_{p=1}^{J}\ \sum_{q=1}^{K} p(x_p = 1)p(z_q = 1)p(\mathbf{s}^l|x_p = 1, z_q = 1)}$$

$$= \frac{\upsilon_j\ \tau_k N(\mathbf{s}^l|\mu_{j,k}, \sigma_{j,k})}{\sum_{p=1}^{J}\sum_{q=1}^{K}[\upsilon_p\tau_q N(\mathbf{s}^l|\mu_{p,q}, \sigma_{p,q})]} \quad (14)$$

The posterior probability value $\pi^l_{(x_j,z_k)}$ can be viewed as the *responsibility* that component (x_j, z_k) takes for explaining observation $\mathbf{s}^l$. We find out this measure of responsibility for each observation in our data set $\overline{\mathbf{S}}$.

5.2 M-step

The M-step of the algorithm corresponds to maximizing the likelihood of the observed data. This leads us to re-estimating the parameters for the next iteration based on the posterior probabilities calculated in the expectation step of the algorithm.

$$\upsilon_j = \frac{\sum_{l=1}^{M} \sum_k \pi^l_{(x_j,z_k)}}{M} \tag{15}$$

$$\tau_k = \frac{\sum_{l=1}^{M} \sum_j \pi^l_{(x_j,z_k)}}{M} \tag{16}$$

$$\mu_{i\ (j,k)} = \frac{\sum_{l=1}^{M} \pi^l_{(x_j,z_k)} s_i^l}{N_{j,k}} \tag{17}$$

where we have defined

$$N_{j,k} = \sum_{l=1}^{M} \pi^l_{(x_j,z_k)} \tag{18}$$

The variance parameter can also be updated accordingly.

5.3 Convergence of Log Likelihood

Each update of the parameters resulting from an E-step followed by an M-step is guaranteed to increase the log likelihood function. The algorithm is deemed to have converged when the change in the log likelihood function falls below a threshold.

$$\ln p(\overline{\mathbf{S}}|\theta) = \sum_{l=1}^{M} \ln \left\{ \sum_{j=1}^{J} \sum_{k=1}^{K} \upsilon_j \tau_k \mathcal{N}(\mathbf{s}^l | \mu_{j,k}, \sigma_{j,k}) \right\} \tag{19}$$

5.4 Handling Identifiability in our Model

In [4] Bishop et al discuss the problem of *identifiability* associated with assigning P sets of parameters to P components. The problem occurs because there are P! ways of assigning P sets of parameters to P components.

In our case each component can be represented as a (location, power-level) pair. We handle the problem of identifiability as follows :

5.4.1 *Indoor Radio Propagation Model*

The indoor radio propagation model is represented as

$$P_{Rx} = P_0 - 10n \log\left(\frac{d}{d_0}\right) \tag{20}$$

where P_0 is the received signal strength at a distance d_0 from the emitter. P_{Rx} is the signal strength(s_i) seen by receiver for a transmitter located at a distance d away from it. n is a parameter which models the behaviour of the environment. This formula effectively initializes the components representing different locations on the map.

To initialize k components (say) which have a common location but vary in power-level, we make use of the observations made in Section 3.2 which show that the observed signal strength is linearly proportional to the transmission power. Thus, once the formula above gives us the signal value for a specific location, we extrapolate the value linearly to initialize each of the k components for that location

In our experiments, we set n = 2. The corresponding signal strength was used to initialize the means ($\mu_{j,k}$). The standard deviation ($\sigma_{j,k}$) was initialized to 5 (and kept fixed to reduce computation time). As subsequent results show, a value of k = 45 is sufficient to hit a constant average error distance.

5.5 Final Location Estimate

Given a real-time received signal vector $\mathbf{s}^{(obs)}$, we can now find the location with the highest probability. We do this by first finding the probability for each (location, power-level) pair and then marginalizing over the power-levels. Thus the estimated location index is given by j^* where

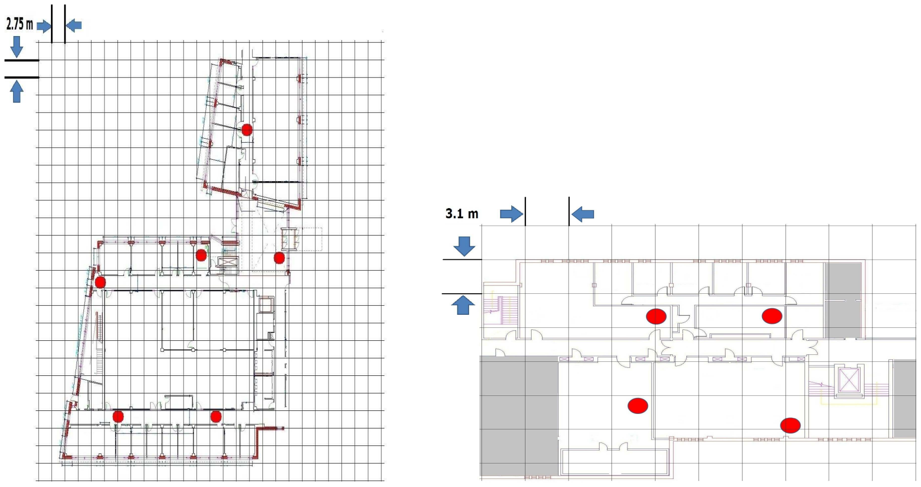
$$j^* = max_j \sum_k P(x_j = 1, z_k = 1 | \mathbf{s}^{(obs)}) \tag{21}$$

6. EXPERIMENT METHODOLOGY

In this section, we describe our methodology for doing the experiments. We start with a description of our system setup. This is followed by an overview of the components of our sniffer devices. We then present details about the two testbeds where we conducted our experiments. Finally we round up this section by discussing the data collection process.

6.1 System Setup

As mentioned briefly in Section 1, our system architecture is along the lines of an infrastructure-based model of location-determination systems. Our system has two main components: stationary sniffer devices in the target space and a centralized server running the GEM algorithm. Sniffers provide overlapping coverage of the target area (similar to how APs are typically deployed inside buildings). The server notifies the sniffers about the mac-id of the target device, the channel number and the listening period. The sniffers then record the signal strength of all packets received that match the server's query. The recorded information is sent to



(a) CEWIT **(b)** CSD

Figure 4: The two testbeds where experiments were conducted

the backed server which makes a location estimation using the GEM algorithm.

In our current prototype, the server communicates with the sniffer devices using the pre-existing in-building power-line ethernet LAN. In the future, our sniffers functionality might be integrated directly into the WLAN APs of a production network. Enterprise APs usually have a centralized controller which can serve as our localization engine. This makes our architecture particularly interesting.

6.2 Sniffer Information

Our sniffer devices are responsible for capturing wireless transmissions made by a Tx-client. We use soekris-net4801 boards as our sniffer devices with atheros-based cm9 cards for wireless captures. Our sniffers are running Pyramid Linux (version 2.6.16-metrix) and we use the default MadWiFi driver which comes with this distribution (0.9.4.5 : svn 1485).

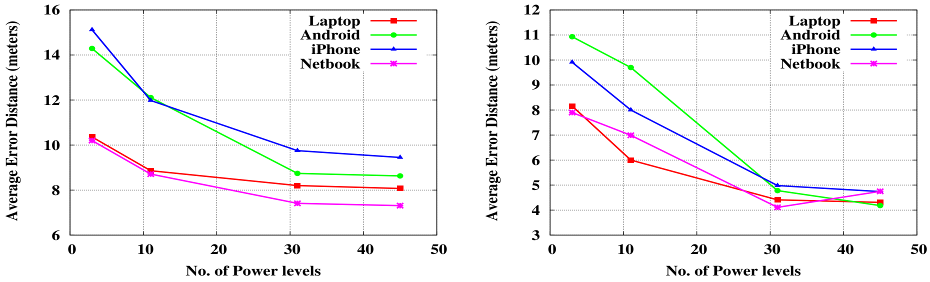
To capture packets we use the Tcpdump software (version 4.0.0 libpcap version 0.9.8) To obtain signal strength information, the MadWiFi driver allows a monitor mode interface to be created and configured with RadioTap header support. From the radio-tap header we can extract the Received Signal strength of each packet received by the sniffer. We verified that the MadWifi driver had a fixed noise-floor in each of our cm9 cards (-95 dbm). In fact the received signal strength of a frame reported by the MadWiFi driver is actually the SNR value (in db) obtained after subtracting the noise-floor from the raw signal strength value. We work directly with the RSSI value (in db) as reported by the driver.

6.3 Testbed Details

We use two different testbeds to Experimentally validate our technique. The first, henceforth called CEWIT, is a large research and educational facility with a dimension of 65 × 50 meter square. The L-shaped floor comprises of several obstructions in the form of concrete walls, glass metal doors, server-rack cabinets housing a host of equipment gear etc. The second, henceforth called CSD, is a portion of the building housing Stony Brook University's Computer Science building. This rectangular-shaped floor has a dimension of 20 × 30 meter square and also contains several obstructions, including concrete walls, wooden partitions etc. Both these testbeds had a continuous flux of people moving around in the building while the experiments were conducted.

6.4 Data Collection Methodology

We perform our experiments with 4 different wireless devices - an android phone, an iphone, a dell laptop and a dell netbook. Our localization is performed on a discretized grid of the target space. The CEWIT testbed is discretized into 45 distinct locations roughly every 5.5 meters. The CSD testbed is divided into 27 distinct locations roughly every 3.3 meters. As part of our location estimation effort, for each of the above four devices we transmit 200 ping packets from every distinct location of the corresponding testbed. This is typically done by having a user hold the mobile device and walk across the floor of the building briefly stopping at each marked location to transmit 200 ping packets. The ground truth is noted at each location before moving on to the new location. Note that the ground truth information is used only for evaluation of the localization error and is not supplied to GEM for training. Each ping packet is separated uniformly apart at a rate of 1 per second. The RSS vector for each transmission is composed of the signal strength value recorded by each individual snif-



(a) CEWIT **(b)** CSD

Figure 5: Avg Error distance as a function of the number of power levels

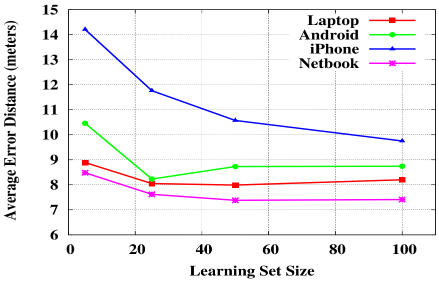


Figure 6: Average Error distance on the CEWIT Dataset as a function of the learning set size

fer. The sequence number in the ping packet is used to form this vector of RSS values from each transmission. Thus, from each distinct location on the map and for each device type, we have a set of 200 RSS tuples. This comprises our entire data set that we use in this paper. Experiments on RADAR [x] and Probabilistic [y] described later in this paper use a subset of this dataset for building the RF signal map and the remainder data for calculating localization error.

In the CEWIT testbed, we have six sniffer devices. The CSD testbed has four sniffers. The circular-dots in Figure 4 show the sniffer positions in each separate testbed. We assume knowledge of the sniffer positions in the map and use this information to calculate the signal strength values given by the indoor radio propagation model (Section 5.4.1). These values are used to initialize our algorithm as explained in Section 5.4 .

7. EVALUATION

In this section, we present a comprehensive overview of our experimental results. We evaluate the performance of GEM on our two experimental testbeds. We attempt to answer the following questions :

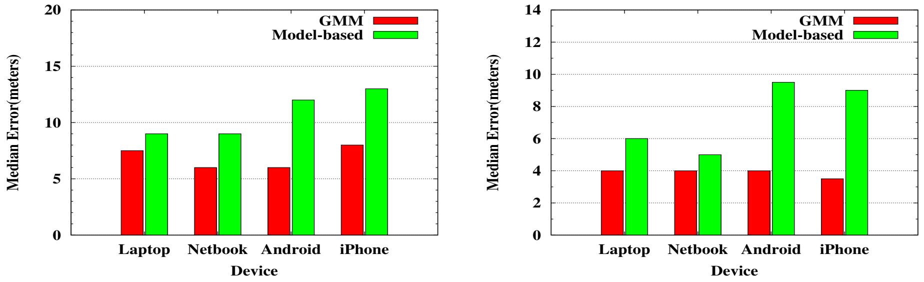
- What is the number of power-levels that we should use in GEM i.e what is the value of k (mentioned in Section p above) that we should use when we run GEM on the back end localization server.
- How does the localization accuracy vary as the size of the learning data-set increases.
- How does GEM perform with respect to a model-based scheme that uses the indoor radio path loss propagation model. This presents a true head-to-head comparison because both the techniques do not need pre-deployment effort and can work on the same granularity of discretization of the target space.
- How does GEM perform with respect to schemes that build RF signal maps like RADAR and Probabilistic. This experiment shows how the WiFi hardware variance problem can impact the accuracy of RF signal map schemes and also show the impact of training granularity for signal map based schemes.
- We also study how the mobility of a client can actually improve GEM's localization accuracy.

7.1 Number of powers levels to use in GEM

As mentioned in Section 6.4 above, the CEWIT testbed has 45 distinct locations and CSD has 27 distinct locations, and for each distinct location on the map and for each device type we have a set of 200 RSS tuples. We divide these 200 tuples into two sets of 100 tuples each: one for learning the GEM parameters and the other for testing the GEM localization results. Each device type is considered separately. Figure 5 shows the results of the average error distance (in meters) for the four devices across varying number of power levels used in GEM. We see that the average error distance hits a plateau after k = 31. This is an interesting result because it helps us bound the number of power levels to use. We use a value of k = 45 in the subsequent experiments.

7.2 Localization accuracy as a function of the learning set size in GEM

Having fixed the number of power levels to use, we now study how the size of the learning data-set changes



(a) CEWIT **(b)** CSD

Figure 7: Baseline Comparisons

the average error distance. Recollect here that as part of our data collection methodology, we have 200 RSS tuples for every location on the map for each of the four device types. This time we again divide the 200 tuples into two sets : one set for learning and the other for testing. The test set size is kept fixed at 100 RSS tuples. From the remaining tuples, the learning set size is varied from 2 tuples going up to 100 tuples. Each device type is considered separately. Figure 6 shows the results of the average error distance (in meters) in the CEWIT testbed as the size of the learning set varies. We observe that for all the four devices, the average error does not vary much as the as we move from 50 training samples to 100 training samples. The CSD testbed results (not included here) converged after 25 training samples itself. The experiments which follow have been done keeping the GEM learning set size at 100 and using the remaining 100 samples for testing the localization accuracy.

7.3 Baseline Comparison with a model-based scheme

Here we analyze the performance of GEM with respect to a model-based scheme that uses the indoor radio path loss propagation model (Section 5.4.1). This presents a true head-to-head comparison, because both the techniques do not need pre-deployment effort and can give a location estimate at the same granularity of discretization of the target space. Both our testbeds, CEWIT and CSD, have been discretized as shown in Figure 4. There are 267 grid vertices in the CEWIT testbed and 36 grid vertices in the CSD testbed. As mentioned in Section 6.4, the data for our experimental evaluation is coming from 45 distinct locations on the CEWIT testbed and 27 distinct locations in the CSD testbed. There are 200 RSS tuples for every location on the map for each of the four device types. As mentioned above in Section aa, GEM is using a learning set size of 100 RSS samples with 45 power levels to build the model. Thus the test-set for both the algorithms is remaining 100 RSS tuples from each location. Each device type is evaluated separately.

The log-distance path loss (LDPL) mentioned in Section 5.4.1 is used to estimate the RSS that should be observed at a sniffer for each grid vertex inside the target-space. These RSS values are used to initialize GEM, as mentioned in section 5.4. The Model-based algorithm also uses these same RSS values with a suitable metric to give a final location estimate. Similar to [x], the model-based algorithm that we use here uses nearest neighbor in signal space (NNSS) as the metric of choice.

Figure 7 shows the median error for both techniques. We see that GEM performs better than the model-based scheme across all device types in both the testbeds.

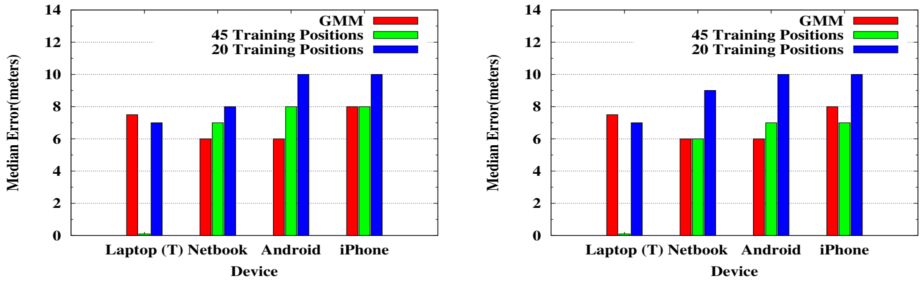
7.4 Comparisons with schemes that use RF signal maps

We now compare the performance of GEM again two schemes that spend considerable pre-deployment effort in first building an RF signal map from RSS signatures collected from various locations inside the target space. Both schemes differ in the way they handle an incoming signature to provide a location estimate.

- RADAR [x] is a deterministic scheme and uses the nearest neighbor in signal space (or an average of k-nearest neighbors) as the metric to give a location estimate.

- Probabilistic [y] on the other hand maintains a probability distribution of the RSS value from various locations, as observed at each sniffer . For the incoming signature, a probability distribution is built over the location space and a maximum likelihood estimate is used to determine position.

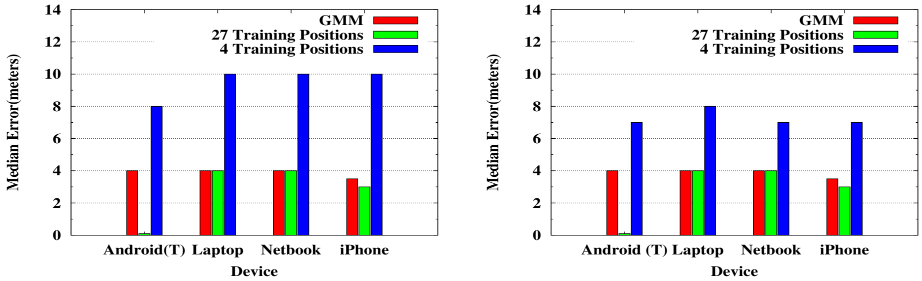
For all three techniques that we compare here viz GEM, RADAR and Probabilistic, we consider the best location as our location estimate. For ease of evaluation, we do not consider the weighted average of the top few locations for any of the three techniques compared here.



(a) Probabilistic

(b) RADAR

Figure 8: Comparisons on the CEWIT testbed



(a) Probabilistic

(b) RADAR

Figure 9: Comparisons on the CSD testbed

To show the WiFi hardware variance problem mentioned in Section 1, we use different devices for location estimation. For the CEWIT testbed, a DELL Laptop was used to train the radio map during the *'offline' phase.* In the CSD testbed, an android phone was used to build the radio map. Based on the radio maps, four different device types are used to give location estimates.

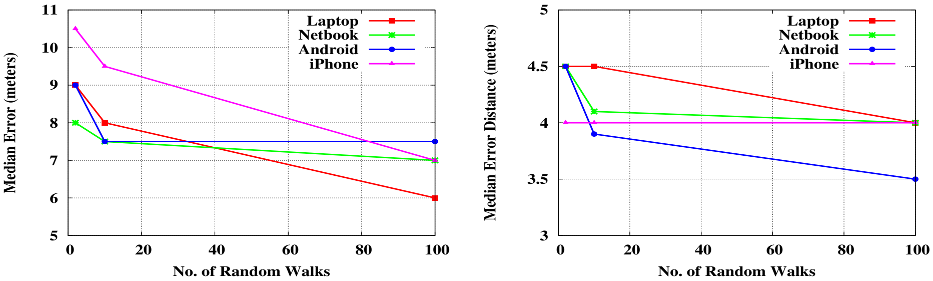
As mentioned in Section 6.4, the data for our experimental evaluation is coming from 45 distinct locations on the CEWIT testbed and 27 distinct locations in the CSD testbed. To understand the effect of the granularity of training locations during the *'offline' training phase* we consider two scenarios. In the first scenario, the RF signal map is built from every distinct location where data was collected, e.g on the CEWIT testbed, the signal map is built from the same 45 distinct locations where the position estimation is being done. In the second scenario, the RF signal map is built from only a subset of the test locations. In the CEWIT testbed, the subset comprises of 20 locations (out of the 45 possible) roughly every 10.3 meters apart. In the CSD testbed, the subset comprises of 4 locations (out of the 27 possible) roughly every 12.7 meters apart. GEM on the other hand requires no training effort. For GEM we use the same discretization of the target space that we use in section 7.3

There are 200 RSS tuples for every location on the map for each of the four device types. We divide these 200 tuples into two disjoint sets of 100 tuples each. The first set is used by RADAR and Probabilistic to build their RF signal maps while GEM uses it as the learning data to build the localization model. The second set of 100 tuples from each location is used as the test data for all three algorithms. Every device type is evaluated separately.

Figure 8 and Figure 9 shows the median error comparison between the three techniques . We make some interesting observations here:

- Hardware variance is a major issue for both RADAR and Probabilistic. When the same device is used for training and testing, the median error is zero. For other devices it jumps up dramatically. This is a critical problem for such techniques because a real-world deployment would be typically unaware of the device type being localized. In fact, we see GEM performing as good as RADAR and Probabilistic for other device types. This is particularly promising because unlike RADAR and Probabilistic, GEM did not have the overhead of a pre-deployment effort.

- When the granularity of sampling is reduced, RADAR and Probabilistic start showing substantially poorer



(a) CEWIT (b) CSD

Figure 10: Mobility

accuracy estimates. Thus location estimates for such techniques are tightly bound to the granularity of the training effort. This makes them unattractive for performing localization in large target spaces. A heavy pre-deployment effort also make these techniques difficult to maintain and update in a dynamic environment.

7.5 Impact of Mobility on GEM's localization accuracy

In this experiment we try to understand how the mobility of a client can effect the location estimates made by GEM. To create the test data set, a user initially walks across all distinct location on the map, making 100 ping transmissions from each location. This forms our test data set which is basically a union of 100 RSS tuples from each distinct location on the map. Now we try to observe the effect of client mobility in localizing this test data set. To do this, the user follows a random walk scheme. In each random walk, the user visits every distinct location in the building floor once, and makes a single ping transmission from that location. This serves as the learning data-set for GEM, in order to build a model for the device and give location estimates for the test set.

We evaluate this scheme on both testbeds across four different devices. Figure 10 shows how the median error of the test data set varies as the mobility (i.e the number of random walks) increases. We observe that mobility actually helps GEM localization accuracy.

8. CONCLUSIONS

In this work, we have developed a server-side technique to localize a wireless client in an indoor environment based on the signal strength parameter of its transmitted packets. We developed a learning-based algorithm that can learn the parameters of the model dynamically from packets captured by the stationary sniffers / APs inside the building. By using dynamic packet captures for parameter estimation, we can provide location estimates which are much more robust in the face of time varying phenomena like movement of people inside the building, opening closing of doors etc. Moreover, this technique can be used on a host of heterogeneous devices operating at different power levels.

We do not have an explicit *training* phase in our technique. Infact, we showed that we can achieve accuracy that is at par with state-of-the-art techniques that use training to build RF-signal maps first. Thus, our technique not only eliminates the intensive time-consuming (often manual) training phase but also makes our technique scalable for large target spaces.

9. PLACEHOLDER. BIBREF. (TO REMOVE)

Haeberlen04 [12], Gwon04 [11], Elnahraway04 [8], Moraes06 [7], Youssef08 [21], Ferris07 [10], Berna03 [2], Lim10 [15], Tsui09 [20], Chintalapudi10 [6], Ladd02 [14], Youssef03 [22], Tao03 [19], Krishnan04 [13], Borman [5], Bilmes97 [3], Roos02 [18], Elnahrawy [9], Bahl00 [1], Molkdar91 [16], Bishop [4], Reynolds [17]

10. REFERENCES

[1] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *in Proceedings of IEEE INFOCOM*, pages 775–784, 2000.

[2] M. Berna, B. Sellner, B. Lisien, S. Thrun, G. Gordon, and F. Pfenning. A learning algorithm for localizing people based on wireless signal strength that uses labeled and unlabeled data. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 1427–1428, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[3] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1997.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] S. Borman. The expectation maximization algorithm: A short tutorial. Technical report.

[6] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 173–184, New York, NY, USA, 2010. ACM.

[7] L. F. M. de Moraes and B. A. A. Nunes. Calibration-free wlan location system based on dynamic mapping of signal strength. In *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, MobiWac '06, pages 92–99, New York, NY, USA, 2006. ACM.

[8] E. Elnahraway, X. Li, and R. P. Martin. The limits of localization using rss. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 283–284, New York, NY, USA, 2004. ACM.

[9] E. Elnahrawy, R. P. Martin, W. hua Ju, P. Krishnan, and D. Madigan. Bayesian indoor positioning systems. In *in Proceedings of IEEE INFOCOM*, pages 1217–1227, 2005.

[10] B. Ferris, D. Fox, and N. Lawrence. Wifi-slam using gaussian process latent variable models. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2480–2485, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[11] Y. Gwon and R. Jain. Error characteristics and calibration-free techniques for wireless lan-based location estimation. In *Proceedings of the second international workshop on Mobility management &amp; wireless access protocols*, MobiWac '04, pages 2–9, New York, NY, USA, 2004. ACM.

[12] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom '04, pages 70–84, New York, NY, USA, 2004. ACM.

[13] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu. A system for lease: Location estimation assisted by stationary emitters for indoor rf wireless networks. In *in Proceedings of IEEE INFOCOM*, pages 1001–1011, 2004.

[14] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless ethernet. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, MobiCom '02, pages 227–238, New York, NY, USA, 2002. ACM.

[15] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo. Zero-configuration indoor localization over ieee 802.11 wireless infrastructure. *Wirel. Netw.*, 16:405–420, February 2010.

[16] D. Molkdar. Review on radio propagation into and within buildings. In *Microwaves, Antennas and Propagation, IEE Proceedings H*, pages 61–73, 1991.

[17] D. Reynolds. Gaussian mixture models. Technical report.

[18] T. Roos, P. Myllymki, H. Tirri, P. Misikangas, and J. Sievnen. A probabilistic approach to wlan user location estimation. In *International Journal of Wireless Information Networks*, pages 155–164, 2002.

[19] P. Tao, A. Rudys, A. M. Ladd, and D. S. Wallach. Wireless lan location-sensing for security applications. In *Proceedings of the 2nd ACM workshop on Wireless security*, WiSe '03, pages 11–20, New York, NY, USA, 2003. ACM.

[20] A. W. Tsui, Y.-H. Chuang, and H.-H. Chu. Unsupervised learning for solving rss hardware variance problem in wifi localization. *Mob. Netw. Appl.*, 14:677–691, October 2009.

[21] M. Youssef and A. Agrawala. The horus location determination system. *Wirel. Netw.*, 14:357–374, June 2008.

[22] M. A. Youssef, A. Agrawala, and A. U. Shankar. Wlan location determination via clustering and probability distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, PERCOM '03, pages 143–, Washington, DC, USA, 2003. IEEE Computer Society.