Network-side Positioning of Cellular-band Devices with Minimal Effort

Ayon Chakraborty, Luis E. Ortiz and Samir R. Das
Computer Science Department, Stony Brook University, Stony Brook, New York 11794, U.S.A.
{aychakrabort, leortiz, samir}@cs.stonybrook.edu

***Abstract*—We address the problem of network-side localization where cellular operators are interested in localizing cellular devices by means of signal strength measurements alone. While fingerprinting-based approaches have been used recently to address this problem, they require significant amount of geo-tagged ('labeled') measurement data that is expensive for the operator to collect. Our goal is to use semi-supervised and unsupervised machine learning techniques to reduce or eliminate this effort without compromising the accuracy of localization. Our experimental results in a university campus (6 sq. km) demonstrate that sub-100m median localization accuracy is achievable with very little or no labeled data so long as enough training is possible with 'unlabeled' measurements. This provides an opportunity for the operator to improve the model with time. We present extensive analysis of the error characteristics to gain insight and improve performance, including understanding spatial properties and developing confidence measures.**

I. Introduction

There is an increasing interest in localizing cellular-band devices on the part of cellular operators. The motivation includes, e.g., location-specific usage patterns for estimating future growth, debugging signal coverage issues or network fault, developing better handoff techniques, and helping antennas to beamform in the right direction. This *network-side* localization is different from more conventional *client-side* localization in the sense that cellular operators typically do not have access to the on-board sensors (e.g., GPS, accelerometer, compass) or other radios (e.g., WiFi, Bluetooth) on the user device that a client-side localization technique can benefit from [12], [4], [13], [18], [24], [27]. Also, many cellular-band devices (e.g., M2M) do not even have any such on-board sensor or other radios while their number is increasing at a very first pace in the cellular networks [20]. Network-side localization is also often *passive* in the sense that there is typically no local software or app resident on the client device. The involvement of the device is limited to sending usual *cellular measurement reports* (signal strength measurements from all or a subset of neighboring base stations[1] that is already part of standard cellular protocols [17]. This is the only information that the operators can exploit for localization.

Indeed, the research community has addressed this form of localization for a long time and generally has reached an agreement that a *fingerprinting* (or *pattern matching* in some literature) based approach [3], [23], [9] provides the most promise, as opposed to a propagation model-based approach.[2] A fingerprinting approach generally has a training phase, where the area in question is mapped by collecting vectors of radio signal features (e.g., signal strengths from all neighboring base stations) labeled by GPS coordinates. At the time of localization, this collection is used as a reference and the vector of features collected from the test device is used to estimate the most probable location. There are a range of deterministic and probabilistic techniques to achieve this with varying degrees of sophistication [22]. However, they all require use of 'labeled' data for training (i.e., training data labeled by GPS or other ground-truth location) – typically more data achieving better accuracy. *Collecting a sufficient amount of 'labeled' data, however, requires effort on the part of the operator.* This could in fact be a cost center, requiring i) either drive tests to fingerprint an area with sufficient granularity or ii) arranging opportunistic collections directly with the subscriber or indirectly via an OTT (over-the-top) service provider.[3]

The goal of this paper is to develop techniques such that *the effort related to supplying labeled data for training is minimized without compromising the final accuracy of the location estimate.* While this 'effort issue' has been recognized in several papers before, the existing papers do not address the problem from the point of view of the cellular operators. They are more interested in client-side localization with active participation of the client device. They use WiFi as opposed to cellular signals thus taking advantage of much denser deployment to provide accuracy. Most importantly, they make liberal use of on-board sensors (e.g., accelerometer or GPS) on the phone for calibration [18], [27], [4], [24], [25] or exploit characteristics only possible in very dense deployments [12]. More related to our work are techniques that use unsupervised or semi-supervised learning techniques [16], [7], [28], [19], [26]. However, they either do not address cellular-specific challenges or do not study the effort question directly.

[1] For simplicity, we will use the term 'base station' to indicate a macro-cell with a unique cell-id.

[2] In a propagation model-based approach, measured radio signal features at the device – such as signal strength, angle of arrival and time difference of arrival, etc – are used to compute range and/or angle measures with respect to the base stations. They in turn are used to compute the mobile location. While such techniques have long been considered (see, e.g., [22]) they suffer from propagation effects (e.g., multipath) thus limiting the accuracy.

[3] For example, a subscriber can be enticed to provide labeled data to the cellular operator via a custom app in exchange of a discount. Also, an OTT provider may already collect location data and business arrangements can be made to acquire these.

978-1-4799-8381-0/15/$31.00 ©2015 IEEE

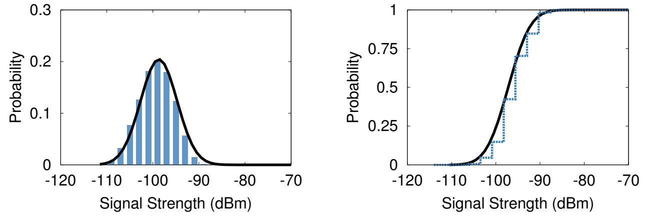
By leveraging well-known machine learning tools, we showcase a technique that provides very good performance with very little labeled data. The technique even works reasonably well with ‘zero’ labeled data. The technique uses the Guassian mixture model (GMM) to model the signal strength vectors heard at the cellular device with its location as a hidden variable (Section II). An EM (expectation maximization) approach is used to learn a distribution over locations that in turn is used to compute a location estimate (Section III). We show if the EM can be initialized ‘right’ – via using a limited amount of labeled data (semi-supervised) or using out-of-band information such as a propagation model (unsupervised) – excellent accuracies can be achieved even with minimal effort (Section IV, VI). The EM is able to leverage ‘unlabeled’ training data that operators can collect without any significant effort. Table I in page 6 highlights the performance of the proposed technique vis-a-vis published literature on network-side cellular localization.

II. Modeling Approach

In this section, our goal is to describe our model and its underlying assumptions. Before we get into the specifics, it is important to point out the fundamental tradeoff in machine learning [15] between ‘goodness-of-fit’ on the training data (how well the model ‘explains’ the training data) and ‘model complexity’ as it relates to our ultimate goal, ‘generalization performance’ (how well the model perform on previously unseen test data, i.e., localization accuracy in our case). In general, overly complex models tend to fit almost any training data well; but, in fact, often too well, leading to overfitting of the model parameters, and poor generalization performance. Conversely, overly simple models tend to do a poor job of fitting the training data, leading to ‘underfitting’ model parameters, and, once again, poor generalization. We have made an attempt to strike the right balance – avoiding unneeded, overly complex model choices and focusing on a practical deployable design.

A. Probabilistic Model

We first discretize the space over which we want to localize by simply overlaying a square grid on the corresponding region.[4] We have used a 15m × 15m grid in our empirical study, but the general modeling approach makes no assumption about how to discretize the space of interest. In particular, the discretization does even not need to be uniform. Now, it is important to understand that finer-grain discretization may help to improve accuracy, but perhaps only up to a point. From a machine-learning perspective, a finer-gain discretization increases model complexity. Statistically, such an increase in model complexity tends to overfit model parameters when learning on the same data as that used for a smaller, less complex model produced from a coarser discretization. Overfitting leads to poor localization accuracy, unless we correspondingly increase the amount of training data.



(a) A Gaussian PDF fitted to the signal

(b) CDF of the fitted Gaussian and the signal

Fig. 1: Validating the Gaussian assumption. The signal strength distribution at a given location is multivariate Gaussian. A specific example location shown using long-term data.

Let L be a discrete random variable representing the different (discrete) locations. We let L take values $l \in \{1, \ldots N\}$, where N is the number of locations. We associate a probability mass function (pmf) p_L to L defined in terms of N-dimensional vector of parameters $\pi \equiv (\pi_1, \ldots, \pi_N)$ corresponding to $p_L(l) \equiv \pi_l$, the (a priori) probability that the test device is at location l (i.e., the event $L = l$). To each location l, we associate a center coordinate (x_l, y_l) in the physical 2-d plane. Also, conditioned on the event that the test mobile is at location $L = l$, let $S \equiv (S_1, \ldots, S_n)$ be an n-dimensional multivariate Gaussian random variable $\mathcal{N}(\mu_l, \Sigma_l)$ with n-dimensional mean parameters μ_l and $n \times n$ covariance-matrix parameter Σ_l modeling the signal strengths that could be measured by the device at location l for each nearby base station. The Gaussian assumption is common for modeling RF signals in similar situations. [5] Figure 1 shows the Gaussian ‘fit’ for long term signal strength data collected at a single location for one base station.

We index the base stations using $j \in \{1, \ldots, n\}$. We let the conditional probability density function (pdf) be the multivariate Gaussian pdf with parameters (μ_l, Σ_l): [6]

$$f_{S|L}(s|l) \equiv \mathcal{N}(s \mid \mu_l, \Sigma_l) \ .$$

B. Naive Bayes Assumption

We also make the assumption that $S_1, \ldots, S_n$ are *conditionally* independent given the event that the device is at location $L = l$. This *naive Bayes assumption* is reasonable as signal strengths of different base stations are generally independent at any given location. [7] This implies that each covariance-matrix parameter Σ_l is actually a diagonal matrix:

$$\Sigma_l \equiv \mathrm{diag}(\sigma^2_{1|l}, \ldots, \sigma^2_{n|l}),$$

[4] The final location estimate can still be in continuous space. This will be clear later.

[5] Note that, from a machine-learning perspective, even if the Gaussian assumption does not strictly hold – so long as it ‘close enough’ – the simplicity of the Gaussian pdf produces much less complex models relative to what the “true” pdf of the signal will induce. This is related to the fundamental tradeoff we mentioned earlier this section.

[6] Note the slight abuse of notation, typical in statistical machine learning, of the symbol $\mathcal{N}$ to correspond to the actual Gaussian pdf.

[7] We acknowledge that this assumption may not be strictly true always. But again this goes back to the model complexity tradeoff mentioned earlier. With this assumption the number of parameters decreases from quadratic to linear in n reducing the model complexity. In addition, it is well-known within the machine-learning community that using the naive Bayes assumption can work really well *even in cases where it clearly does not hold* [6].

with conditional variances $\sigma^2_{i|l} \equiv \mathrm{var}\,(S_i \mid L = l)$, for each i; that is, the conditional covariance between any pair of two different random variables (S_i, S_j) with $i \neq j$ is $\mathrm{cov}\,(S_i, S_j) = 0$. Thus, the conditional pdf of S given $L = l$ simplifies to

$$f_{S|L}(s|l) \equiv \prod_{i=1}^{n} \mathcal{N}(s_i \mid \mu_{i|l}, \sigma^2_{i|l}) \ ,$$

where [8]

$$\mathcal{N}(s_i \mid \mu_{i|l}, \sigma^2_{i|l}) \equiv \frac{1}{\sqrt{2\,\pi}\sigma_{i|l}} \exp\left(-\frac{1}{2} \frac{(s_i - \mu_{i|l})^2}{\sigma^2_{i|l}} \right) \ ,$$

is the 1-d Gaussian pdf.

Putting everything together, we can state the joint (mixed discrete-continuous) probability function associated with our statistical model as:

$$f_{L,S}(l, s) \equiv p_L(l)\ f_{S|L}(s|l) \equiv \pi_l \prod_{i=1}^{n} \mathcal{N}(s_i \mid \mu_{i|l}, \sigma^2_{i|l}) \ .$$

For simplicity, denote the model parameters by $\theta \equiv (\pi, \mu, \sigma^2)$, where μ and σ^2 correspond to all conditional means and variance parameters.

When L is a *hidden* (i.e., unknown) random variable, the *marginal pdf* of S is a *mixture of independent Gaussians*, *aka*, a *naive Bayes Gaussian mixture model (GMM)*:

$$\begin{aligned} f^{\theta}_S(s) &\equiv \sum_{l=1}^{N} p_L(l)\ f_{S|L}(s|l) \\ &\equiv \sum_{l=1}^{N} \pi_l \prod_{i=1}^{n} \mathcal{N}(s_i \mid \mu_{i|l}, \sigma^2_{i|l}) \ . \end{aligned}$$

C. Localization using Naive Bayes Gaussian Models

Given model parameters θ and an input signal strength vector s at the test device, we localize the device using the posterior (conditional) probability over the location L given $S = s$. In particular, we first compute the *posterior* pmf of L given $S = s$ as

$$p^{\theta}_{L|S}(l|s) \equiv \frac{\pi_l \prod_{i=1}^{n} \mathcal{N}(s_i \mid \mu_{i|l}, \sigma^2_{i|l})}{\sum_{l'=1}^{N} \pi_{l'} \prod_{i=1}^{n} \mathcal{N}(s_i \mid \mu_{i|l'}, \sigma^2_{i|l'})} \ ,$$

for all l. Then, we use *Bayesian averaging* to estimate the location as (x^*, y^*), where

$$x^* = \sum_{l} p^{\theta}_{L|S}(l|s) x_l; \ \ y^* = \sum_{l} p^{\theta}_{L|S}(l|s) y_l.$$

Note that this estimated location is no longer discrete. [Other possibilities such as using a maximum a posteriori (MAP) estimate exist; that is, use (x_{l^*}, y_{l^*}) where $l^* \in \arg\max_l p^{\theta}_{L|S}(l|s)$. But full Bayesian averaging has worked better for us.]

In order to apply this technique, we need to learn the model parameters $\theta \equiv (\pi, \mu, \sigma^2)$. We do this via semi-supervised and unsupervised learning techniques described in the following section.

III. Semi-supervised Learning of Model Parameters

We use maximum-likelihood estimation (MLE), as commonly applied in statistical machine learning, to learn the model parameters θ from a given dataset of measurement samples (also called examples). Suppose we have collected a dataset $D = \{d^{(1)}, d^{(2)}, \ldots, d^{(m)}\}$ of m measurement samples of signal strength vectors at various locations, and further assume that the samples are independent, identically distributed (i.i.d.). We use k to index the samples in D. We split the samples in D into two datasets D^{train} and D^{test} for training and testing, respectively. Without loss of generality, because we can always permute the order of the samples, uniformly at random, assume $D^{\text{train}} \equiv \{d^{(1)}, d^{(2)}, \ldots, d^{(m^{\text{train}})}\}$ corresponds to the first m^{train} samples in D, and assign the remaining samples to D^{test}. Only a subset of the samples in D^{train} are assumed to be have location labels. They are the only ones that are assumed to contribute to effort or cost.

A. Identifiability Problem and Initialization

If none of the samples in the D^{train} dataset has a location label, they are assumed *hidden* and the setting is completely *unsupervised*. In that case, our model fully becomes a naive Bayes GMM. Each training sample $d^{(k)} = (., s^{(k)})$ would consist of the measured signal strengths only, with the label missing. (The period indicates the missing label.)

The problem with learning in this setting is *identifiability*: data alone is insufficient to distinguish between the different locations! Because we are in a completely unsupervised setting, the likelihood function $\mathcal{L}_{D^{\text{train}}}(\theta) \equiv \prod_{k=1}^{m^{\text{train}}} f^{\theta}_S(s^{(k)})$ is the same for any permutation of the numbers, $\{1, \ldots, N\}$, associated with the locations. There is no way to map these (now abstract) location labels to physical space.

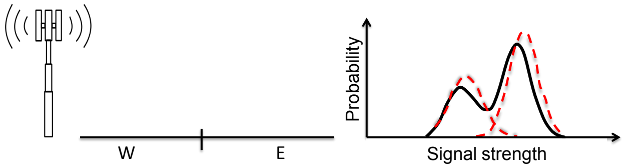


Fig. 2: Illustrating the identifiability problem: (Left) Base station and two locations. (Right) Signal strength distribution shown in solid line – comprising of two Gaussians (dashed lines) for the two locations.

We illustrate this problem using an 1-d example in Figure 2. In this setting there is one base station and two locations: W and E. Our abstract model would have $N = 2$, so that L can take values in $\{1, 2\}$. In essence, at this point we do not know whether *abstract* location 1 would be associated to *actual* location W or E; and similarly for abstract location 2. Assume that the samples in D^{train} exhibit the shown signal strength distribution that has two Gaussian components corresponding to the two locations. Since no labeled samples are available, the combined signal strength distribution (solid line) is the only information we have. While an MLE algorithm

[8] Note that π here corresponds to the constant $3.142\ldots$, not the parameters of the pmf of L.

is powerful enough to estimate whether a given test sample is more likely from one of the Gaussians versus the other, it cannot tell whether *that* Gaussian represents E or W. To see this, suppose that we would like to associate abstract location 1 to W and abstract location 2 to E. Suppose that the parameters that lead to the correct mapping are $\theta = (\pi, \mu, \sigma^2)$ so that $\pi_1 = \pi^E, \pi_2 = \pi^W, \mu_{1|1} = \mu^E, \mu_{1|2} = \mu^W, \sigma^2_{1|1} = (\sigma^E)^2, \sigma^2_{1|2} = (\sigma^W)^2$. Now let $\theta' = (\pi', \mu', (\sigma^2)')$ be a permutation of θ over locations L leading to the incorrect mapping: $\pi'_1 = \pi^W, \pi'_2 = \pi^E, \mu'_{1|1} = \mu^W, \mu'_{1|2} = \mu^E, (\sigma^2_{1|1})' = (\sigma^W)^2, (\sigma^2_{1|2})' = (\sigma^E)^2$. Then, because $\mathcal{L}_{D^{\text{train}}}(\theta) = \mathcal{L}_{D^{\text{train}}}(\theta')$ (i.e., both θ and θ' explain the training data equally well), one cannot determine whether the MLE algorithm outputs the correct mapping from abstract to actual physical locations.

To address this, we need to provide some form of 'guidance' to the MLE algorithm so that such mapping to physical space is possible. One way to provide such guidance is via an appropriate *initialization*. That is, we initialize the parameters of the model in such a way that the MLE algorithm converges to the "right" areas of the likelihood function $\mathcal{L}_{D^{\text{train}}}(\theta)$ in parameter space, the set of all possible values of θ (i.e., those likely to lead to the correct mapping from abstract location labels to the actual physical space). Continuing with Figure 2 as an example, we would like guide the MLE algorithm to output θ, not θ', despite both set of parameters having the same likelihood value. We explore two ways to achieve such initialization.

1. *Semi-supervised approach* – Assume that there is a 'small' number of labeled samples in the training set D^{train}. These samples coupled with some additional modeling (e.g., spatial interpolation - to be discussed later) can very roughly estimate model parameters μ and σ^2 at every location.

2. *Unsupervised approach* – Exploit an 'out-of-band' model to roughly estimate the initial model parameters μ and σ^2 at every location. This can be done using propagation models using the knowledge of the location and transmission characteristics of the base stations and terrain features.

B. Learning Algorithm: Expectation Maximization

We now describe the version of the expectation-maximization (EM) algorithm [5] we use for semi-supervised learning based on MLE in our model. Note that the unsupervised and supervised versions are just special cases and do not need a separate description. Recall that labeled examples have the form $d^{(k)} = (l^{(k)}, s^{(k)})$, while unlabeled examples have the form $d^{(k)} = (., s^{(k)})$, where the period indicates the missing label. The unsupervised version does not have any labeled example.

The first step is initializing parameters $\theta \equiv (\pi, \mu, \sigma^2)$. We set θ to $\theta^{(0)}$ where the superscript denotes steps in the algorithm. The initialization follows one of the two approaches described in the previous subsection. We describe the specifics in the later two sections, Section IV for semi-supervised and Section VI for unsupervised.

Then, for $t = 0, 1, 2, 3, \ldots$, where t denotes steps:

1) *Expectation step:* for all samples k in D^{train}, and all locations l, compute the *posterior* probability, with respect to the model induced by parameters $\theta^{(t)}$, that the location-region $L = l$ given the data sample $d^{(k)}$: [9]

$$\gamma_{l|k} \equiv \mathbf{P}\left(L = l \mid d^{(k)}; \theta^{(t)}\right) = \begin{cases} \mathbf{1}\left[l = l^{(k)}\right], & \text{if sample } k \text{ is labeled,} \\ p^{\theta^{(t)}}_{L|S}(l|s^{(k)}), & \text{otherwise.} \end{cases}$$

Then, set $m_l^{\text{train}} \leftarrow \sum_k \gamma_{l|k}$.

2) *Maximization step:*

$$\pi_l^{(t+1)} \leftarrow \frac{m_l^{\text{train}}}{m^{\text{train}}}, \quad \mu_{i|l}^{(t+1)} \leftarrow \frac{1}{m_l^{\text{train}}} \sum_k \gamma_{l|k} s_i^{(k)},$$

$$(\sigma^2_{i|l})^{(t+1)} \leftarrow \frac{1}{m_l^{\text{train}}} \sum_k \gamma_{l|k} (s_i^{(k)} - \mu_{i|l}^{(t+1)})^2 .$$

3) *Update parameters:*

$$\theta^{(t+1)} \leftarrow (\pi^{(t+1)}, \mu^{(t+1)}, (\sigma^2)^{(t+1)}) .$$

Of course, at some point the process must stop, and there exist many different stopping rules (e.g., based on the absolute or relative change in likelihood of the parameters, or change in the values of the parameters themselves). In the case of supervised learning, in which all the training examples are labeled, it is not hard to see that the value of the parameters do not change after the first iteration; thus, in that case, we can stop after one iteration. In general, for both semi-supervised and unsupervised implementations we use in this paper, we stop the process when the log of the posterior probabilities, $p^{\theta^{(t)}}_{L|S}(l|s^{(k)})$, do not change by more than 1%.

IV. Evaluation

This section is devoted to a baseline performance analysis of the semi-supervised approach. We first describe the data set, followed by the specifics of the initialization and then the results.

A. Data Set

We have used Nexus 5 phones running Android (version 4.4.2) for collecting measurement data. We developed a mobile app that logs the signal strength measurements on device from the serving as well as neighboring base stations along with the timestamps and GPS coordinates. We deployed the application on several phones and collected measurements across a university campus (approx 6 sq. kms). Only outdoor measurements are collected in navigable areas – campus roads, walkways, parking areas, etc. The signal strengths are obtained using appropriate Android APIs in terms of ASU that takes integer values from 0 to 31. It is later converted to dBm using the formula $(-113 + 2 * ASU)$ [10]. The app automatically logs the data whenever there is a change in signal strength or location. Though the measurement set up and our methodology

[9] The term $\mathbf{1}\left[l = l^{(k)}\right]$ evaluates to 1 if the data sample $d^{(k)} = (l^{(k)}, s^{(k)})$ is labeled and the label is $l = l^{(k)}$; and to 0 otherwise.

[10] 3GPP TS 27.007 version 8.5.0 Release 8, http://bit.ly/1ttLMYs

are radio agnostic, we specifically did this study for GSM only.The reason for this much denser GSM coverage at our location relative to 3/4G. A single cellular provider is used.

The measurement data is collected on random days over a month resulting in a collection of about 40K data points. A total of 9 base stations are observed. Thus, the feature vector S has 9 dimensions ($n = 9$). But at a given location only a subset of base stations is actually heard. For a cell that cannot be heard at a location we 'impute' it to -113 dBm, the noise floor. The measurements are done somewhat opportunistically in the sense that no specific attempt was made to sample the space uniformly. Thus, there are denser data in some areas as opposed to others – roughly reflecting popularity of various campus locations.

B. Initialization

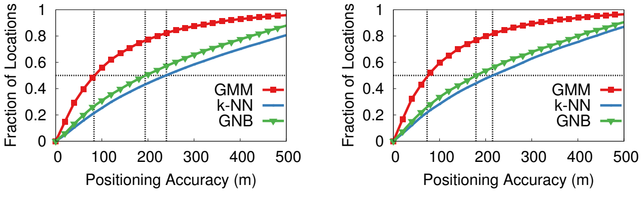
We discretize the navigable outdoor space on the campus[11] by overlaying a 15m × 15m grid. Each grid cell represents a candidate location L. Overall, there are about 3K such locations ($N = 3$K). The 40K sample dataset is split into two parts for training and testing (D^{train} and D^{test}) with the testing size $m^{\text{test}} = 15$K. Various training set sizes (m^{train}) are chosen (up to 25K) for different evaluations. *The data in D^{train} is assumed without location labels except that a small fraction is assumed to have labels when semi-supervised approach is used.* For the unsupervised approach (Section VI), no label is assumed. For ease of presentation, we concentrate only on the semi-supervised approach here.

Following the discussion in Section III-A, the model parameters $\theta \equiv (\pi, \mu, \sigma^2)$ are initialized as follows. For each location L_l where there is at least one labeled example in D^{train}, the mean is computed directly and is used as the initial value ($\mu_l^{(0)}$). If there is no labeled example in L_l, spatial interpolation (using other labeled examples) is used to determine the mean. Here, we benefit from recent studies on RF mapping using sparse sampling [2]. While various interpolation techniques are possible, recent studies [2] have indicated that even very simple linear, distance-weighted techniques can do reasonably well. We use such a technique called IDW or inverse distance weighting [21].

Variance is computed globally (and not per location) for each cell separately, and is used for initialization ($(\sigma_l^2)^{(0)}$). The priors π_l are initialized uniformly over all locations because no other information is available.

C. Results

After initialization the model is learnt using the EM algorithm in Section III-B. Then the model is used to compute the location estimate (x^*, y^*) for each test sample $d^{(k)}$ in D^{test} as described in Section II. The *localization error* (sometimes we call this *localization accuracy* or plain *accuracy*) is then computed as the physical distance between the actual location of the test sample $l^{(k)}$ and the estimated location (x^*, y^*).



(a) 1% Labeled data (b) 10% Labeled data.

Fig. 3: Baseline performance comparison with supervised learning techniques.

The get a good insight on the error we compare the performance of the proposed approach with two baseline *fingerprinting*-based approaches. In the latter, any sample used for training must be labeled. This comparison is important in our problem set up for two reasons. First, it is conceivable that the absolute error is probably not the right measure of real performance. The error fundamentally depends on the terrain and how base stations are deployed, their density, etc.[12] Second, such a relative comparison showcases why supervised approaches are not well suited for the specific network-side localization problem we are considering.

Figure 3 shows the CDF of errors for the proposed GMM-based technique along with two standard fingerprinting-based approaches – i) 'supervised' Gaussian Naive Bayes (GNB): this is equivalent to running just one iteration of the EM approach presented in Section III using only the labeled data; and ii) k-Nearest Neighbors (k-NN): a test sample is localized to the location of the majority of k nearest neighbors in the signal space ($k = 10$ is used).

For fairness, the amount of labeled examples are assumed to be the same in all cases. Among the 25K samples available for training, two cases are considered – assuming 1% and 10% of the training samples as labeled. The GNB and k-NN techniques can only utilize these labeled samples for training. GMM on the other hand, additionally benefits from the remaining (unlabeled) samples in the training set to improve learning. *Note that median performance of GMM is almost a factor of 2.5 better than the others.* GMM provides a median error of ≈90 m (1% labels) or ≈70 m (10% labels) while the supervised techniques provide ≈200 m.

At this point it is also instructive to summarize the overall performance vs effort in our work vis-a-vis published literature on network-side cellular localization. Table I provides such a summary. Note the overall excellent performance of proposed semi-supervised and unsupervised techniques (to be described in Section VI) while incurring minimal effort.

[11] Non-navigable areas are not considered as there is no data point in these areas in our study.

[12] Generally speaking higher density is expected to reduce error. We will explore this further in SectionVI. As a quick example of how density might play a role, many recent papers on WiFi localization boast of location errors within a few meters [12], [4], [13], [18], [24], [27] while most cellular localization papers present close to 2 orders of magnitude higher errors [3], [23], [8], [11]. Our testing area located in a suburb does not have cellular coverage equivalent to a population hotspot. Also moving forward, with advent of small cells, increasing density and hence better absolute error is to be expected.

No.	Technique	Median error (m)	Scenario	Effort
1	k-NN	277 ([3]), 255 ([8])	Metro (residential)	Training Set: 4350 km drive trace Testing Set: 89 km ([3]), 38 km ([8]) trace
2	k-NN	94 ([3]), 100 ([23])	Metro (downtown)	Training Set: 4350 km drive trace Testing Set: 24 km ([3],[8]) trace
3	k-NN	293 ([8])	Metro (suburban)	Training Set: 4350 km drive trace Testing Set: 51 km ([8]) trace
4	k-NN	177-221 [11]	Campus wide (urban)	*Unspecified*
5	Gaussian process	196 ([3]), 208 ([8])	Metro (residential)	Same as 1
6	Gaussian Process	126 ([3]), 128 ([8])	Metro (downtown)	Same as 2
7	Gaussian process	236 ([8])	Metro (suburban)	Same as 3
8	Particle Filter	155 ([16])	Residential Area	25% data used for calibraton
9	Supervised GNB, k-NN	200-250 [this paper, sec IV]	Campus wide (suburban)	1% ($\approx$ 50 points/sq. km)
10	Semi-supervised GMM	90 [this paper, sec IV,V]	Same as 9	Same as 9
11	Unsupervised GMM	90-120 [this paper, sec VI]	Same as 9	Negligible

TABLE I: Accuracy vs effort tradeoffs in various localization schemes.

V. Analysis of Errors

It is clear from the CDF in Figure 3 that the errors are well-distributed over a wide range, while the median error is ≈70-90 m, the 90 percentile error is much larger, ≈350 m. Our general goal in this section is to explore possible dependencies of the error on components of our techniques or specifics of the input data. Such understanding can help improve performance.

In various analyses here we will vary the amount of labeled data or training data or both. Unless otherwise specified, assume that we use all the training data (25K), out of which 2% (500) are labeled and rest are unlabeled. The error bars in all the plots indicate the 25th and 75th percentiles.

A. Spatial distribution of errors

Figure 4 shows the distribution of errors overlaid on the map. Note that there is noticeable spatial clustering of similar errors, specifically for larger errors. This is expected as the RF signal strength is modulated by the terrain characteristics. Understanding this phenomenon well can improve performance, e.g., more labeled data can be collected from such locations if appropriate.

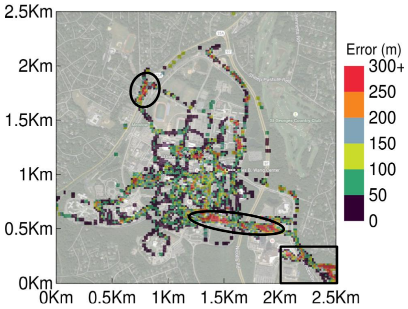


Fig. 4: Spatial distribution of errors with clusters of large error regions circled.

To quantify the spatial nature of the errors we show their spatial autocorrelation property in terms of 'Moran's I'. Moran's I [14] is a commonly used measure of spatial autocorrelation. A concept of distance is used to indicate proximity and is used as 'weights.' Moran's I is defined as:

$$I = \frac{N_o}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij}(x_i - \overline{x})(x_j - \overline{x})}{\sum_i (x_i - \overline{x})^2}, \tag{1}$$

where x is the random variable studied (in this case the error) $\overline{x}$ being the sample mean, x_i's are the observations. w_{ij} is the weight associated with each pair (x_i, x_j) and N_o is the number of observations. In our case, x_i is median error in location i, and w_{ij} is the physical distance between locations i and j.

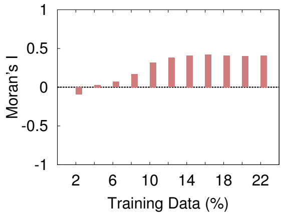


Fig. 5: Moran's I vs amount of training data.

Figure 5 plots Moran's I with varying training data size (with labeled data size fixed at 500). Note that Moran's I increases first with increasing training data (D^{train}) size, but then it converges around ≈ 0.4. This plot has a few takeaways. First, significant amount of spatial auto-correlation is present in general[13] Second, the correlation does not change with increasing training data beyond point. This indicates that this correlation is a fundamental property of the data and may not go away even if more training data is supplied in specific locations/regions.

B. Impact of initialization

Recall from Section IV-B that the locations that do not have any labeled examples use interpolation for initializing the mean μ. It is possible that such interpolations contribute to errors for these locations. To study this, we analyze the distribution of errors in the 'interpolated' locations vs 'non-interpolated' locations. Figure 6 shows the difference between the median localization errors in these two groups plotted with increasing amount of labeled data (with training data size fixed). Note that with very little labeled data, there are significantly larger errors for the interpolated locations. However, the interpolated locations behave increasingly similarly to their non-interpolated counterparts when the amount of labeled data

[13] Moran's I can range from -1 to +1; -1 indicates perfect dispersion, 0 means random, +1 indicates perfect correlation.

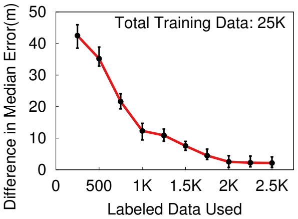


Fig. 6: Errors at interpolated locations versus actual measurement locations.

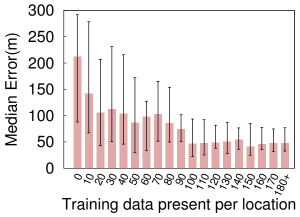


Fig. 7: Dependence of error with amount of training data per location.

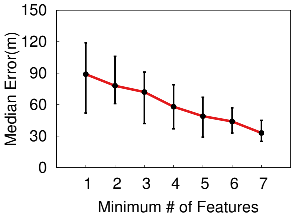


Fig. 8: Dependence of error on minimum number of features.

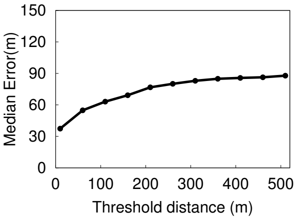


Fig. 9: Confidence on error.

is increased and they are very similar with $> 7\%$ labeled data. There are a few takeaways here: i) Interpolation performance improves with more and denser sampling – this is intuitive. ii) Interpolation become almost perfect with even with a very small amount of labeled data ($> 7\%$), while there is room for improvement when labeled data is very few ($< 2\%$). Note that we have used a very simple interpolation. More sophisticated methods may improve performance in these extreme situations. We leave this investigation for future work.

A related question is whether more training data (D^{train}) in locations with poor accuracy would help. In that case, one can improve accuracy by collecting more training data from regions with poor accuracy. To explore this, we explore the relationship of errors with the amount of training data in each location. See Figure 7 where errors are classified according to the amount of training data in each location. While very little training produces larger errors we do see some form of convergence beyond about 100 training samples with median error going down to $\approx 50\,\text{m}$. The latter value likely characterizes the baseline noise in the system that more training cannot help mitigate.

C. Impact of cellular density

We have seen in Section V-A that errors exhibit some amount of spatial correlation. We now are interested in understanding what location specific parameter is causing this behavior. The usual suspect here is the number of features, i.e., the number of neighboring base stations (including the serving base station) the mobile can hear. These are the number of measured components in the signal strength vector S (the rest of the components are imputed as described in Section IV-A).

To study this we create 7 separate data sets. In each data set, we use a minimum threshold on the number of features (1 through 7). The rest of the data is discarded for that set. The entire learning and testing process is repeated for each set with training and test data drawn from that set only. Figure 8 shows that accuracy improves steadily with the lower bound on the number of features that are available in the data set. With ≥ 7 features the accuracy is quite good, $\approx 35\,\text{m}$. But for the other extreme – where all data is considered (≥ 1 feature) – the accuracy is $\approx 90\,\text{m}$. Also, the variation of error is much less with more features. Generally, this is a good news as denser urban locations frequented by more people is expected to have dense cellular coverage as opposed to suburban/rural areas with lower population density. From the operators' perspective – given the type of applications they consider – more populous areas also need more accuracy. We expect an even better accuracy for future denser deployments (e.g., small cells).

D. Confidence of Estimation

Until now we mostly analyzed how errors behave. From a practical application's perspective, it is useful to produce a *confidence measure* that relates to the estimation error. Thus, when a measured test sample s is produced for localization, we want to produce a confidence measure along with the location estimate. This measure enables the application to treat different confidence levels differently, e.g., the application may simply choose to discard lower confidence estimates acting as if it did not even receive any measurement.

We have explored several possible confidence measures, but not all seem to bear the essential property that the measure should have a fair degree of correlation with error. That is, higher confidence should also produce smaller error with higher probability. If it does not, such a measure is not very useful to the application. The measure that has worked well works as follows. Given a test sample s, we take the top two posterior probability components $p^{\theta}_{L|S}(l|s)$ and determine the Euclidean distance between the corresponding two locations l. This distance is used as the confidence.[14] Intuitively, the top probabilities should be in the vicinity of each other. If the top two probabilities are spatially apart then the system must be confused about the location. Figure 9 plots error vs confidence. For this plot the confidence is 'thresholded,' i.e., the error for a confidence value, say $90\,\text{m}$, includes all errors where the confidence is *at most* $90\,\text{m}$. Note that the median error increases with poorer confidence (i.e., larger distance) leveling off after certain point.

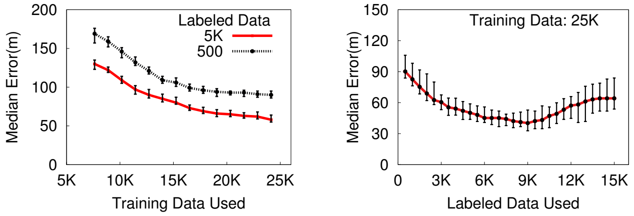
E. Learning Curve

Recall from Section I, the measurement effort (cost) spent in localization is the amount of labeled data. The amount of 'unlabeled' training data do not present any effort as they can be collected virtually for free and could even be a part of the routine measurements. Thus, it is interesting to analyze how – given a fixed budget (amount of labeled data) – increasing the amount of (unlabeled) training data could

[14] Other measures that we explored but did not work well include: i) difference between the top two probabilities, ii) area of the convex hull containing a threshold amount of probability mass.

improve localization accuracy. A use case for such a set up is as follows. The operator collects some (small) amount of labeled data according to its budget. Then with time, more and more unlabeled training data are collected for free and the model is rebuilt as new training data comes along.[15] We plot the median error with the amount of training data given a fixed amount of labeled data showcasing how such a system would learn over time. See this 'learning curve' in Figure 10(a). Note the gradual improvement of error with time (more training data). Eventually such a curve is expected to level off when the performance reaches the system noise. However, we do not have enough data to reach that point in this study.

Finally, Figure 10(b) highlights performance vs cost where the cost is in terms of the amount of labeled data. The training data size is constant. Note how cost impacts performance. There is only minor improvement beyond ≈3K labeled data. Very small amount of labeled data (≈500) still provides acceptable performance. Note also the increase in errors with a very large amount of labeled data. This is due to overfitting as the training size is fixed.



(a) Increasing training set

(b) Increasing labeled data set

Fig. 10: Learning Curve

VI. Unsupervised Approach

In this section, we explore the unsupervised (zero-effort) approach. Recall from Section III-A an unsupervised approach suffers from the identifiability problem in that it is unable to distinguish between the $N!$ possible numbering of the N locations as all of these choices provide identical solutions for the EM. A semi-supervised approach as discussed above addresses this issue by initializing the EM closer to the final solution by supplying a small amount of labeled samples. Without access to any labeled samples an unsupervised approach must make use of 'out-of-band' model/information to do this initialization. Such a model could be a radio propagation model (or other form of RF mapping tools, e.g., [1]) used by the cellular operators for deployment or coverage studies. Broadly, given the location of a base station, transmission characteristics and knowledge of terrain, such tools are able to estimate signal strengths at various locations. These models could work without any form of calibration (true zero-effort).

Since we do not have access to such tools and do not have knowledge of the transmission characteristics of the base station (e.g., transmit power and antenna parameters) we take the following approach: We first find the location of the base stations via manual search. We then use the standard log-distance path loss with log-normal shadowing model and fit parameters using a small subset of the labeled data we already have.[16] This provides us with a useful propagation model.[17]

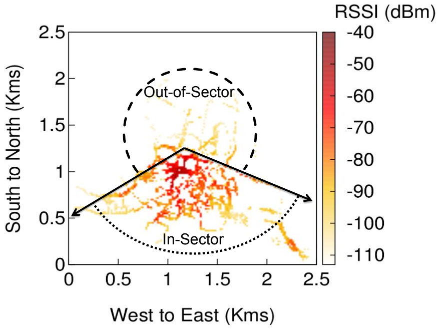
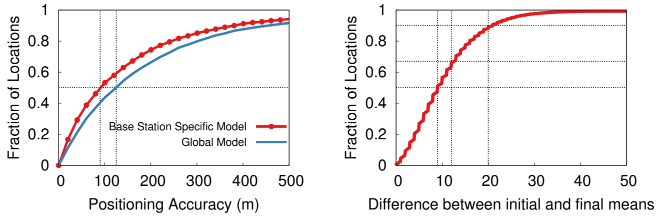


Fig. 11: Service sector of an antenna. The In-Sector and Out-of-Sector locations have different propagation models. Actual signal strength measurements are shown.

We use two different models – i) global model using aggregated data, ii) base station-specific models using base station-specific data. Note that the base station-specific model takes into account the terrain characteristics more closely. Each model takes into account the antenna pattern. We empirically validate that the transmitter beamwidth to be 120°.[18] We then empirically estimate the orientation of the sector by following the signal strength gradients. Then we develop two separate models per transmitter - *in-sector* and *out-of-sector*, corresponding to all locations that fall within the sector and outside the sector. See Figure 11 for an illustration. Finally, for all models we force a minimum value of the signal power ($-113\,$dBm, equal to the noise floor).



(a) Comparing two propagation models

(b) Difference between initial and final means(dB) for the base station-specific model.

Fig. 12: Performance of unsupervised approach with two different propagation models.

[15] It is possible that the model can actually be updated incrementally, though we did not do it.

[16] From the log-distance path loss model with log-normal shadowing is as follows: $RSS(d) = RSS(d_0) - 10\alpha\log_{10}\left(\frac{d}{d_0}\right) + \chi_\sigma$, where $RSS(d)$ is the measured signal at distance d, d_0 is a reference distance from the tower location, α is the path-loss exponent and χ_σ is a shadowing component represented by a Gaussian random variable with zero mean and std. deviation σ. Ignoring the shadowing component, $\log_{10}(d)$ and $RSS(d)$ should have a linear relationship. We use a linear regression to find the value of α.

[17] A reader may think that this approach is not true zero-effort, since labeled data is being used. However, we want to stress is that all we need a good enough initialization with a reasonable propagation model. Operators routinely use such modeling. Without access to it, however, we must create one. We are using labeled samples to create this model to proxy for one that the operators should already have.

[18] From empirical observations, the signal strengths pattern appears to be formed by a 120° sector antenna.

Computationally, the unsupervised approach is similar to the semi-supervised except that (i) during initialization, the signal means (μ_l) are estimated using the propagation model and (ii) no labeled data is assumed in the EM. Figure 12(a) shows the localization accuracy for the global and base station-specific propagation models. Obviously, the global model being more generic performs somewhat poorly. Note that the median localization accuracy is quite competitive with semi-supervised technique (Figure 3). The global model produces a median error of ≈120 m while the base station-specific model gives a median error of ≈90 m. Contrast this with ≈70 m or 90 m for semi-supervised and much worse ≈200 m for supervised. Also see Table I in page 6. Figure 12(b) shows the power of EM, showing the statistics of the difference between the initial (from propagation model) and final (after EM converges) signal means. The median difference is large – as much as ≈10 db showing that the propagation model is not 'solving the problem' for EM even for the base station-specific model's case. Final estimates are quite far from the initial. The initialization is only helping solve the identifiability issue.

VII. Related Works

Cellular localization is a topic that has been explored for a long time. For brevity, we only review related works that *focus specifically on network-side localization* and do not make use of short-range radios such as WiFi or Bluetooth or on-board sensors such as GPS, compass or accelerometer. As mentioned in Section I using them provides a very different problem setting.

The survey paper by Sun et al. [22] covers a set of commonly used techniques for network-side localization. Relevant to our work, a set of paper solve the localization problem using statistical machine learning tools [16], [28], [19], [26]. A series of studies by Youssef et al. [10], [9], Chen et al. [3],Varshavsky et al. [23] explores localization using GSM-based signals at metropolitan scales. While some of these studies show competitive performance as ours, they do not focus on reduction of effort as a design goal. See also Table I in page 6.

VIII. Conclusion

The paper demonstrates a significant opportunity of employing semi- and unsupervised learning in addressing the network-side localization problem. We have shown sub-100 m accuracy (median) with either technique, while using minimal or zero labeled samples so long as enough training data is available. But since such training does not need labeled examples, it is virtually free for the operator. Our analysis further highlights that when the base station density is high, the accuracy improves to about 30 m (median). This presents a significant opportunity in future small cell deployments, providing a potential of achieving GPS-like performance but still with minimal effort in labeled data collection.

Acknowledgement

This work is partially supported by a gift from Futurewei Technologies, Inc. and NSF grant CNS-1117719.

References

[1] Wireless EM propagation software. http://www.remcom.com/wireless-insite.

[2] A. Achtzehn, J. Riihijarvi, and P. Mahonen. Improving accuracy for TVWS geolocation databases: Results from measurement-driven estimation approaches. In *Proceedings IEEE DySPAN*, 2014.

[3] M. Y. Chen et al. Practical metropolitan-scale positioning for GSM phones. In *Proceedings of ACM UbiComp*, 2006.

[4] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of ACM MobiCom*, 2010.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977.

[6] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[7] A. Goswami, L. E. Ortiz, and S. R. Das. WiGEM: A learning-based approach for indoor localization. In *Proceedings of ACM CoNEXT*, 2011.

[8] B. Hahnel, F. Dirk, and D. Fox. Gaussian processes for signal strength-based location estimation. In *Proceeding of Robotics: Science and Systems*, 2006.

[9] M. Ibrahim and M. Youssef. A hidden markov model for localization using low-end gsm cell phones. In *Proceedings of IEEE ICC*, 2011.

[10] M. Ibrahim and M. Youssef. Cellsense: An accurate energy-efficient gsm positioning system. *Vehicular Technology, IEEE Transactions on*, 61(1):286–296, 2012.

[11] M. Ibrahim and M. Youssef. Enabling wide deployment of GSM localization over heterogeneous phones. In *Proceeding of ICC*, 2013.

[12] H. Lim et al. Zero-configuration, robust indoor localization: Theory and experimentation. In *Proceedings of IEEE Infocom*, 2006.

[13] E. Martin et al. Precise indoor localization using smart phones. In *Proceedings of ACM Multimedia*, 2010.

[14] P. A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37:1733, 1950.

[15] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, MA, 2012.

[16] P. Nurmi, S. Bhattacharya, and J. Kukkonen. A grid-based algorithm for on-device gsm positioning. In *Proceedings of ACM UbiComp*, 2010.

[17] M. Rahnema. *UMTS network planning, optimization, and inter-operation with GSM*. Wiley-IEEE Press, 2008.

[18] A. Rai et al. Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of ACM MobiCom*, 2012.

[19] A. Schwaighofer et al. GPPS: a gaussian process positioning system for cellular networks. In *Proceedings of NIPS*, 2003.

[20] M. Z. Shafiq et al. A first look at cellular machine-to-machine traffic: large scale measurement and characterization. In *Proceedings of ACM SIGMETRICS*, 2012.

[21] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of 23rd ACM National Conference*, 1968.

[22] G. Sun et al. Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs. *IEEE Signal Processing Magazine*, 22(4):12–23, 2005.

[23] A. Varshavsky et al. Are GSM phones THE solution for localization? In *Proceedings of MCSA, IEEE Workshop on*, 2006.

[24] H. Wang et al. No need to war-drive: unsupervised indoor localization. In *Proceedings of ACM MobiSys*, 2012.

[25] C. Wu et al. WILL: Wireless indoor localization without site survey. *Parallel and Distributed Systems, IEEE Transactions on*, 24(4):839–848, 2013.

[26] Z.-l. Wu et al. Location estimation via support vector regression. *Mobile Computing, IEEE Transactions on*, 6(3):311–321, 2007.

[27] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of ACM MobiCom*, 2012.

[28] H. Zang, F. Baccelli, and J. Bolot. Bayesian inference for localization in cellular networks. In *Proceedings of IEEE Infocom*, 2010.