# ALIGNING TRAINING MODELS WITH SMARTPHONE PROPERTIES IN WIFI FINGERPRINTING BASED INDOOR LOCALIZATION

*Manh Kha Hoang, Joerg Schmalenstroeer, Reinhold Haeb-Umbach*

Department of Communications Engineering, University of Paderborn, 33098 Paderborn, Germany
Email: {hoang, schmalen, haeb}@nt.uni-paderborn.de

## ABSTRACT

We are concerned with the so-called fingerprinting method for WiFi-based indoor positioning, where the measured received signal strength index (RSSI) is compared with training data to come up with an estimate of the user's location. We introduce a method for adapting the trained models to the statistics of the RSSI values of the target (testing) WiFi device, which is derived from the Maximum Likelihood Linear Regression (MLLR) framework. By introducing regression classes the assumption of a linear relationship between the RSSI readings of the testing device and the training data is relaxed, leading to superior adaptation performance. Parameter adaptation formulas are derived for the general case of censored and dropped data. While censoring occurs due to the limited sensitivity of WiFi chips, dropping is probably caused by limitations of the operating system of the portable devices. Experiments both on simulated and real-world data demonstrate the effectiveness of the proposed algorithms.

***Index Terms*—** Indoor positioning, signal strength, censored data, expectation maximization, maximum likelihood linear regression

## 1. INTRODUCTION

Due to the unavailability of GPS, accurate indoor positioning remains a challenging task. Many algorithms have been proposed, such as lateration based, angulation based and fingerprinting based techniques using radio signals, such as Ultra-Wideband, ZigBee, Bluetooth Low Energy or WiFi, often in combination with accelerometer and magnetometer measurements, see e.g. [1, 2, 3, 4].

However, due to multipath propagation, missing line-of-sight, unsynchronized clocks etc., lateration and angulation based techniques seem to be unsuitable for low-cost positioning systems. As a result the fingerprinting method, which has been pioneered in [2], is of particular interest, where in an offline (training) phase the readings of the RSSI of access points (APs) measured at the possible target positions are gathered, and the measurements during the online (classification) phase are compared to these training data to come up with a decision on the user's location. WiFi fingerprinting has found widespread use due to the availability of the infrastructure and the ubiquity of WiFi in portable devices, such as laptops, smartphones etc..

Accurate positioning based on RF signal strength faces also many challenges. It is well known that the WiFi scan results can vary greatly, depending on the number of people in the vicinity, the orientation of the device, and many more factors. To overcome these variations, refined statistical models and classification rules have been developed [4, 5, 6].

It was also found that WiFi chips exhibit a limited sensitivity and that RSSI values below the reception threshold (e.g., $-100\,\mathrm{dBm}$) are censored, i.e., can be thought of being clipped to that minimum value. In [5] we have proposed an Expectation Maximization (EM) algorithm to estimate the parameters of such censored GAUSSIAN data. In this way one was able to exploit the location information that is given by the fact that some APs are partly 'inaudible' at certain locations while they are not at other positions.

In [7] it was pointed out that differences in RSSI readings are not only caused by environmental and propagation factors, but that they are also caused by different hardware, i.e., different chipsets or antennas. The authors observed differences as high as $-30\,\mathrm{dBm}$ between chipsets of different vendors and up to $-20\,\mathrm{dBm}$ among chipsets of the same vendor, and even devices using the same chipset could show significantly different RSSI readings.

This and the so-called drop-out problem mentioned in [7] we also experienced in our own investigations. The latter refers to the fact that occasionally RSSI readings of supposedly strong enough access points were not reported by the software. A closer look at this phenomenon revealed that the drop-out rate increased with the number of access points: if more than 20 APs were above the reception threshold, the drop-out rate rose to about 20%, with a tendency to increase to up to 40% for more than 40 audible APs. With the ubiquity of WiFi, such large numbers of APs are not unrealistic. We conjecture that the dropping phenomenon is caused by limitations of the operating system, such as limited buffer sizes or time-outs.

In this paper we propose algorithms to counteract the negative effects of drop-outs and differences in hardware variations on the positioning accuracy. We will extend the EM algorithm for the estimation of the parameters of a GAUSSIAN in the presence of censored data that was presented in [5] to the case of random dropping of measurements at an unknown dropping rate. Furthermore, an adaptation algorithm will be proposed that adapts the parameters of the trained models to the statistics of the RSSI measurements of the target testing device. Compared to the calibration method proposed in [8], where a linear relationship was assumed between the RSSI readings of the training and the target device, that was learnt by a Least Squares approach, we relax the linearity assumption by clustering user positions in multiple regression classes that share the same linear relationship, however different from the other clusters. We also investigate the benefits from adapting the variances, in addition to the means of the training models. The parameters of the transformations are learnt as to maximize the likelihood of the adaptation data, adopting a very successful approach to speaker adaptation in automatic speech recognition, known as Maximum Likelihood Linear Regression (MLLR) [9, 10, 11].

## 2. PARAMETER ESTIMATION AND CLASSIFICATION IN THE PRESENCE OF CENSORED AND DROPPED DATA

### 2.1. Parameter Estimation

In [5], an EM algorithm has been derived for estimating the parameters of censored GAUSSIAN data. In this paper we extend the model to consider the presence of dropped data in the measurements, see Fig. 1.

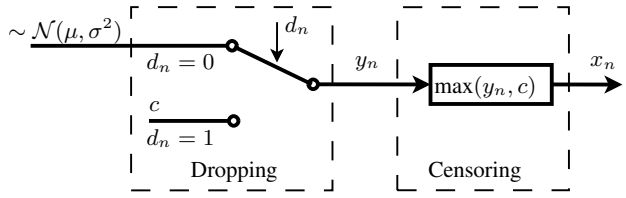


**Fig. 1**. Measurement model.

Here, the hidden variables $d_n$, $n = 1, \dots N$, indicate whether an observation is dropped ($d_n = 1$) or not ($d_n = 0$), where $\pi = P(d_n = 1)$ is called the *dropping rate* in the following. The variables are gathered in the binary vector $\mathbf{d} = [d_1, \dots, d_N]$, where $N$ is the number of measurements. Further, define $\mathbf{y} = [y_1, ..., y_N]$, where the $y_n$ are i.i.d. with GAUSSIAN probability density function (PDF) $p_Y(y_n) = \mathcal{N}(y_n; \mu, \sigma^2)$ if $d_n = 0$, and $y_n = c$ if $d_n = 1$, where we set $c$ to the smallest measurable RSSI value (e.g., $c = -100\,\text{dBm}$). Observable are the censored data $\mathbf{x} = x_1, ..., x_N$, where $x_n = \max(y_n, c)$, with the censoring threshold $c$ as above.

Notice the difference between censoring and drop-outs: Even if the parameters of the GAUSSIAN are such that practically no censoring occurs (because $\mu \gg c$), there will still occur drop-outs!

Our goal is to estimate the parameters $\theta = \{\mu, \sigma^2, \pi\}$. This will be achieved by the Expectation Maximization (EM) algorithm, where the hidden variables are $\{\mathbf{y}, \mathbf{d}\}$ and the observable are $\{\mathbf{x}\}$. Noting that $x_n$ does not convey more information about $\theta$ than $y_n$, the expected log-likelihood of the complete data, given the observed, is given by:

$$\begin{aligned} Q(\theta; \theta^{(\kappa)}) &= E[\ln[p(\mathbf{y}, \mathbf{d}; \theta)] | \mathbf{x}; \theta^{(\kappa)}] \\ &= \sum_{n=1}^{N} \sum_{d_n=0}^{1} \int_{-\infty}^{\infty} \ln[p(y_n, d_n; \theta)] \cdot p(y_n, d_n | x_n; \theta^{(\kappa)}) dy_n, \end{aligned} \quad (1)$$

where $\kappa$ is the iteration index.

We first write $p(y_n, d_n | x_n) = p(y_n | d_n, x_n) P(d_n | x_n)$. Then we find for data that are not dropped ($d_n = 0$)

$$p(y_n | d_n = 0, x_n; \theta^{(\kappa)}) = \begin{cases} \delta(y_n - x_n), \text{ if } x_n > c \\ \frac{\mathcal{N}(y_n; \theta^{(\kappa)})}{I_0(\theta^{(\kappa)})}, \text{ if } x_n = c \end{cases} \quad (2)$$

and for data that are dropped ($d_n = 1$)

$$p(y_n | d_n = 1, x_n; \theta^{(\kappa)}) = \begin{cases} 0, & \text{if } x_n > c \\ \delta(y_n - c), & \text{if } x_n = c \end{cases}. \quad (3)$$

Here, $I_j(\theta^{(\kappa)})$, $j = 0, 1, 2$, is defined as follows:

$$I_j(\theta^{(\kappa)}) = \int_{-\infty}^{c} y^j \mathcal{N}\left(y; \mu^{(\kappa)}, (\sigma^2)^{(\kappa)}\right) dy. \quad (4)$$

Furthermore, the posterior of the hidden variable $d_n$ can be computed using Bayes' rule

$$P(d_n | x_n; \theta^{(\kappa)}) = \frac{p(x_n | d_n; \theta^{(\kappa)}) P(d_n)}{\sum_{d_n=0}^{1} p(x_n | d_n; \theta^{(\kappa)}) P(d_n)}. \quad (5)$$

Four cases can be discerned. Using the notation $\beta_n(d, z) := P(d_n | z(x_n); \theta^{(\kappa)})$, where $z(x_n) = 1$ indicates that $x_n = c$ and $z(x_n) = 0$ that $x_n > c$, we obtain:

$$\beta_n(1, 1) = \frac{\pi^{(\kappa)}}{I_0(\theta^{(\kappa)})(1 - \pi^{(\kappa)}) + \pi^{(\kappa)}} \quad (6)$$

and $\beta_n(0, 1) = 1 - \beta_n(1, 1)$. Furthermore, it is obvious that $\beta_n(1, 0) = 0$ and thus $\beta_n(0, 0) = 1$.

We further note that

$$\begin{aligned} p(y_n, d_n; \theta) &= P(d_n; \theta) p(y_n | d_n; \theta) \\ &= \begin{cases} \pi \delta(y_n - c), & \text{if } d_n = 1 \\ (1 - \pi) \mathcal{N}(y_n; \theta), & \text{if } d_n = 0 \end{cases} \end{aligned} \quad (7)$$

Using all this in (1) and calculating the derivative of the auxiliary function, the following iterative parameter estimation formulas can be derived:

$$\mu^{(\kappa+1)} = \frac{\sum_{n=1}^{N} (1 - z_n) x_n + \frac{I_1(\theta^{(\kappa)})}{I_0(\theta^{(\kappa)})} \sum_{n=1}^{N} z_n \beta_n(0, 1)}{N - \sum_{n=1}^{N} z_n \beta_n(1, 1)} \quad (8)$$

$$\begin{aligned} \left(\sigma^2\right)^{(\kappa+1)} &= \frac{\sum_{n=1}^{N} \left[ z_n \left( \frac{I_2(\theta^{(\kappa)})}{I_0(\theta^{(\kappa)})} - 2\mu \frac{I_1(\theta^{(\kappa)})}{I_0(\theta^{(\kappa)})} + \mu^2 \right) \beta_n(0, 1) \right]}{N - \sum_{n=1}^{N} z_n \beta_n(1, 1)} \\ &\quad + \frac{\sum_{n=1}^{N} \left[ (1 - z_n)(x_n - \mu)^2 \right]}{N - \sum_{n=1}^{N} z_n \beta_n(1, 1)} \end{aligned} \quad (9)$$

$$\pi^{(\kappa+1)} = \frac{\sum_{n=1}^{N} z_n \beta_n(1, 1)}{N}, \quad (10)$$

These formulas reduce to the ones presented in [5] for censored data only, i.e., if the dropping rate is set to $\pi = 0$.

### 2.2. Classification

While the derivation in the last section was done for scalar measurements for simplicity, we now assume that RSSI readings of $N_{AP}$ access points are available, which are gathered in the observation vector $\mathbf{x}$. An optimal classification rule for censored GAUSSIAN data has been given in [5] where the user's location is determined as that position $\ell_k$, which has the highest posterior probability:

$$p(\ell_k | \mathbf{x}) = \frac{\prod_{i=1}^{N_{AP}} p(x_i | \ell_k) P(\ell_k)}{\sum_{k'=1}^{K} \prod_{i=1}^{N_{AP}} p(x_i | \ell_{k'}) P(\ell_{k'})} \quad (11)$$

where $K$ is the number of offline training locations and $x_i$ is the RSSI of $i$-th AP. $P(\ell_k)$ is the prior on the position. Here we assumed independence of the RSSIs of different APs.

For censored and dropped GAUSSIAN data, the likelihood $p(x_i | \ell_k)$ can be calculated as follows

$$p(x_i | \ell_k) = \begin{cases} \mathcal{N}(x_i; \hat{\mu}_{k,i}, \hat{\sigma}^2_{k,i}), & \text{if } x_i > c \\ (1 - \hat{\pi}_{k,i}) I_0(\hat{\mu}_{k,i}, \hat{\sigma}^2_{k,i}) + \hat{\pi}_{k,i}, & \text{if } x_i = c \end{cases}. \quad (12)$$

Here, $(\hat{\mu}_{k,i}, \hat{\sigma}^2_{k,i}, \hat{\pi}_{k,i})$ are the estimated parameters of the $i$-th AP at location $\ell_k$.

## 3. MLLR BASED ADAPTATION

The RSSI readings that are used for positioning have been shown to depend on the hardware, in particular on the WiFi chipset and on the antenna used in the target device that is to be located. This dependency can result in large positioning errors if the target device

employs different hardware than the device that was used to collect the training data.

MLLR adaptation aims at determining the parameters of an adaptation matrix $\mathbf{W}$ such that the likelihood of the adaptation data is maximized [9, 10]. In the following we assume supervised adaptation, i.e., the presence of a calibration phase, in which the positions where the adaptation RSSI values have been measured, are known.

Let $\mu_{k,i}$ be the mean value of the GAUSSIAN describing the PDF of the RSSI values of the $i$-th AP at position $\ell_k$. We gather the mean values of all APs in vectors $\boldsymbol{\mu}_k = [\mu_{k,1}, \ldots, \mu_{k,i}, \ldots, \mu_{k,N_{AP}}]^T$ for all positions $\ell_k$, $k = 1, \ldots, K$. Our goal is to estimate adapted mean vectors $\hat{\boldsymbol{\mu}}_k$ via

$$\hat{\boldsymbol{\mu}}_k = \mathbf{W}^{(c(k))}\boldsymbol{\xi}_k = \mathbf{A}^{(c(k))}\boldsymbol{\mu}_k + \mathbf{b}^{(c(k))}, \tag{13}$$

where $\mathbf{W} = [\mathbf{A}|\mathbf{b}]$ is the $(N_{AP} \times N_{AP} + 1)$-dimensional augmented transformation matrix which gathers the parameters of the affine transform and $\boldsymbol{\xi}_k = (\boldsymbol{\mu}_k^T, 1)^T$ is the extended mean vector. Note that we consider the general case of multiple transformation matrices, where $c(k) \in \{1, C\}$ is the regression class index, which indicates which of the $C$ affine transforms is to be applied to the GAUSSIAN describing position $\ell_k$.

Similarly, let $\boldsymbol{\Sigma}_k$ be the (diagonal) covariance matrix of the GAUSSIAN random vector modelling the RSSI readings of the APs at position $\ell_k$. We will also consider variance adaptation:

$$\hat{\boldsymbol{\Sigma}}_k = \mathbf{B}_k^T \mathbf{H}^{(c(k))} \mathbf{B}_k \tag{14}$$

where $\mathbf{H}^{(c(k))}$ is the transform to be estimated and $\mathbf{B}_k$ is the Choleski factor of $\boldsymbol{\Sigma}_k$. Assuming diagonal covariance matrices, eq. (14) simplifies to $\hat{\sigma}_{k,i}^2 = h_i^{(c(k))}\sigma_{k,i}^2; i = 1, \ldots, N_{AP}$.

To determine the maximum likelihood estimates of the transformation matrices for mean and variance from the adaptation data we employ again the EM algorithm.

Let $\gamma_k(n)$ be the posterior probability that RSSI measurement vector $\mathbf{x}_n$ is from position $\ell_k$. In the case of supervised adaptation considered here it is equal to one if the measurement was indeed from position $\ell_k$ and zero else.

The expected loglikelihood of the complete data, given the observed, has a similar form as before:

$$Q(\lambda; \lambda^{(\kappa)}) = \sum_{k=1}^{K}\sum_{n=1}^{N}\gamma_k(n)\sum_{i=1}^{N_{AP}}\sum_{d_{n,i}=0}^{1} \int_{-\infty}^{\infty} \ln\left(p(y_{n,i}, d_{n,i}; \lambda)\right) p(y_{n,i}, d_{n,i}|x_{n,i}; \lambda^{(\kappa)}) dy_{n,i}. \tag{15}$$

Here $\lambda = \{\lambda_{k,i}; k = 1, \ldots, K; i = 1, \ldots, N_{AP}\}$, where $\lambda_{k,i} = \{\pi_{k,i}, \mathbf{w}_i^{(c(k))}, h_i^{(c(k))}\}$, is a short hand notation for the parameters to be estimated. Here, $\mathbf{w}_i^{(c(k))}$ is the $i$-th row of $\mathbf{W}^{(c(k))}$. In (15) the first sum is over the locations for which adaptation data are available, the second enumerates the adaptation data, while the third is over all APs and the fourth over the possible values of the random variable $d_{n,i}$ which indicates whether the $i$-th component of the $n$-th observation is dropped or not.

The terms in Eq. (15) have already been discussed in Section 2 with the only difference that $y_{n,i}$ is now assumed to be drawn from a GAUSSIAN with adapted parameters: $y_{n,i} \sim \mathcal{N}(\hat{\mu}_{k,i} = \mathbf{w}_i^{(c(k))}\boldsymbol{\xi}_k, \hat{\sigma}_{k,i}^2 = h_i^{(c(k))}\sigma_{k,i}^2)$.

Reestimation formulas for $\pi_{k,i}$, $\mathbf{w}_i^{(c(k))}$ and $h_i^{(c(k))}$ are obtained by computing the derivatives of the auxiliary function in Eq. (15) w.r.t. these parameters and setting them to zero. This leads to the following update equation for the rows of $\mathbf{W}^{(c(k))}$:

$$\sum_{n=1}^{N}\frac{\gamma_k(n)}{\sigma_{k,i}^2}\left(z_{n,i}\frac{I_1(\lambda_{k,i}^{(\kappa)})}{I_0(\lambda_{k,i}^{(\kappa)})}\beta_{k,n,i}(0,1) + (1 - z_{n,i})x_{n,i}\right)\xi_k^T = \sum_{n=1}^{N}\frac{\gamma_k(n)}{\sigma_{k,i}^2}\left(1 - z_{n,i}\beta_{k,n,i}(1,1)\right)\left((\mathbf{w}_i^{(c(k))})^T\right)^{(\kappa+1)}\xi_k\xi_k^T, \tag{16}$$

while explicit updates can be found for $\pi_{k,i}$ and $h_i^{(c(k))}$:

$$(\pi_{k,i})^{(\kappa+1)} = \frac{\sum_{n=1}^{N}\gamma_k(n)z_{n,i}\beta_{k,n,i}(1,1)}{\sum_{n=1}^{N}\gamma_k(n)} \tag{17}$$

$$\left(h_i^{(c(k))}\right)^{(\kappa+1)} = \frac{1}{\sigma_{k,i}^2\sum_{n=1}^{N}\gamma_k(n)\left(1 - z_{n,i}\beta_{k,n,i}(1,1)\right)} \sum_{n=1}^{N}\gamma_k(n)\Bigg[(1 - z_{n,i})(x_{n,i} - \hat{\mu}_{k,i})^2\beta_{k,n,i}(0,0) + z_{n,i}\left(\frac{I_2(\lambda_{k,i}^{(\kappa)})}{I_0(\lambda_{k,i}^{(\kappa)})} - 2\frac{I_1(\lambda_{k,i}^{(\kappa)})}{I_0(\lambda_{k,i}^{(\kappa)})}\hat{\mu}_{k,i} + \hat{\mu}_{k,i}^2\right)\beta_{k,n,i}(0,1)\Bigg]. \tag{18}$$

Here, $\beta_{k,n,i}(d,z) = P(d_{n,i}|z(x_{n,i}), \lambda_{k,i}^{(\kappa)})$, similar as in Section 2.1.

As can be seen in equations (16) and (18), the dropped or censored data also contribute to the estimates, besides the observable ones. If there are no censored and dropped data, then $\beta_{k,n,i}(0,1) = 1$ and $\beta_{k,n,i}(1,1) = 0$, and the equations reduce to the re-estimation formulas for $\mathbf{W}$ and $\mathbf{H}$ given in [11].

## 4. REGRESSION CLASSES

The above formulas have been derived for the general case of multiple transformation matrices. This allows for the user locations to form regression classes, where all positions in the same regression class share the same transformation, while the PDFs corresponding to positions in other regression classes transform differently.

A critical issue is how to define the regression classes, within which the parameters of all positions change in the same manner. Here we assume that locations with similar mean RSSI values transform in the same manner, irrespective of the actual identity of the access points, as long as they operate in the same frequency band. We thus conduct a k-means clustering of the mean vectors $\boldsymbol{\mu}_k$ of the PDFs of all positions $\ell_k$ to obtain $C$ clusters.

## 5. EXPERIMENTAL RESULTS

### 5.1. Classification on Artificial Data

In a first set of experiments we generated artificial data to assess the impact of the drop-out rate and the amount of adaptation data on the classification performance.

We generated 1000 samples of training data for each of $K = 6$ positions and for $N_{AP} = 2$ access points. We then assumed a single affine transformation of the means of the training data for each location according to $\hat{\boldsymbol{\mu}}_k = \mathbf{A}\boldsymbol{\mu}_k + \mathbf{b}$, where $\mathbf{A} = (0.9, 0; 0, 0.9)$ and $\mathbf{b} = [-3, 2]^T$. Adaptation and test data were then generated by sampling from the transformed GAUSSIANs with means $\hat{\boldsymbol{\mu}}_k$ and variances as those of the training data. While the amount of adaptation data was gradually increased from 1% to 5, 10, 50 and 100% of the number of training samples, the number of test samples was fixed at 100 observations per position. We experimented with a drop-out rate of 0% (no drop-outs, $\pi = 0$) and 20% ($\pi = 0.2$).

**Table 1**. Classification results (% positions correctly classified) when training and test data are generated from same parameter set

| Aware of dropping | yes | no |
|---|---|---|
| $\pi = 0$ | 86 | 86 |
| $\pi = 0.2$ | 75 | 64 |

**Table 2**. Adaptation performance (% positions correctly classified) on artifical data as a function of amount of adaptation data

| No. Pos | $\pi$ | 0% | 1% | 5% | 10% | 50% | 100% |
|---|---|---|---|---|---|---|---|
| 3 | 0 | 68 | 81 | 85 | 85 | 86 | 86 |
| | 0.2 | 56 | 70 | 73 | 74 | 75 | 75 |
| 6 | 0 | 68 | 85 | 86 | 86 | 86 | 86 |
| | 0.2 | 56 | 72 | 75 | 75 | 75 | 75 |

Table 1 discusses the impact of the drop-out rate without adaptation. Here, the 'adaptation' data were used for a completely new training from scratch for each of the 6 positions. It can be seen that the parameter estimation formulas derived in Section 2 are able to cope with an unknown drop-out rate, which is estimated (aware of dropping: yes), while assuming absence of drop-outs (aware of dropping: no) leads to clearly inferior classification accuracy if indeed drop-outs occurred.

Table 2 demonstrates the effectiveness of applying the proposed adaptation algorithms. We can notice that when the number of adaptation data per position is small, i.e., less than 10%, classification results using 6 positions with adaptation data are about 1 to 4% better than those when adaptation data are only available for 3 positions. The column with 0% of adaptation data shows the results when no adaptation is performed. Comparing with the results of Table 1 it is clear that the classification results after adaptation approach those of retraining, except when there is only 1% of adaptation data, even if adaptation data are only available for 3 out of 6 positions.

### 5.2. Classification on Field Data

We conducted measurements on 3 floors of an office building with roughly 30 rooms (lecture halls, office and laboratory rooms), where each floor has an overall size of 35 m by 35 m, RSSI values were taken at 60 different positions with an average distance of 5.0 m between 2 positions. 200 measurements were taken per position with 2 different smartphones at each position. The data of the first smartphone was used to estimate the training models while data from the second smartphone was divided into 2 sets at each position, the first being used for adaptation (0, 5, 25, or 75 samples) or retraining (190 samples) from scratch and the second for testing (10 samples). For the measured data, the estimated dropping rate averaged over all APs and all positions was approximately 0.3.

For the adaptation procedure, we sorted the estimated training means at each position in descending order, and used only the 8 strongest APs to estimate the adaptation matrices since the contribution of the remaining APs to the likelihood was negligible. The adaptation matrices are then used to calculate the adapted parameters of the 8 strongest APs at all positions being in the same cluster.

Table 3 shows the dependency of the positioning accuracy on the number of locations for which adaptation data are available and the amount of adaptation data. The results in Table 3 are the average of the root mean square (RMS) positioning error of 50 experiments. In each experiment, test data, adaptation data and the positions with adaptation data are randomly selected. As expected, the more positions there are with adaptation data and the more adaptation samples per positions, the better the positioning results.

**Table 3**. RMS positioning error (in [m]) as a function of the amount of positions having adaptation data and the amount of adatation data

| Condition | | Adaptation Method | | |
|---|---|---|---|---|
| No. Pos with adapt. data | No. adapt. data | $\mu$ & $\sigma^2$, **A** full | $\mu$ only, **A** full | $\mu$ only, **A** diag |
| | 0 | 6.15 | | |
| 15 | 5 | 5.08 | 4.76 | 3.93 |
| | 25 | 4.60 | 4.41 | 3.93 |
| | 75 | 4.62 | 4.52 | 3.93 |
| 30 | 5 | 3.86 | 3.96 | 3.84 |
| | 25 | 3.82 | 3.96 | 3.85 |
| | 75 | 3.76 | 3.91 | 3.85 |
| 60 | 5 | 3.74 | 3.93 | 3.84 |
| | 25 | 3.67 | 3.87 | 3.83 |
| | 75 | 3.65 | 3.87 | 3.84 |
| | retrain | 2.43 | | |

**Table 4**. Effect of number of clusters on RMS positioning error

| No. of clusters | 1 | 2 | 3 |
|---|---|---|---|
| RMS pos. error [m] | 3.76 | 3.48 | 3.25 |

For all considered adaptation methods, improvements in positioning accuracy are obtained, even when very few adaptation data available. However, mean and variance adaptation with a completely filled matrix **A** delivers only the best results if there are sufficiently many adaptation data, while the use of only mean adaptation with a diagonal **A** is superior if only few adaptation data are available, since fewer parameters need to be estimated.

However, positioning accuracy when all 60 positions have adaptation data is still well below the accuracy achievable, when training and test data are collected by the same smartphone, which is 2.43 m.

Next we measured the impact of multiple regression classes. Table 4 shows the positioning results when doing clustering before (mean and variance) adaptation, assuming that in each cluster 50% of positions have adaptation data with 75 adaptation samples per position. It has to be noted that the optimal number of regression classes depends on the amount of adaptation data available. The more adaptation data the more transformation matrices can be reliably estimated. In our setup the best results were achieved with 3 clusters, which led to a reduction of the RMS positioning error from 3.76 to 3.25 m.

## 6. CONCLUSIONS

In this paper we have developed a method for adapting trained models of RSSI measurements to the specifics of the testing WiFi device within the MLLR framework. Parameter estimation formulas have been derived both for mean and variance adaption. They account for the facts that part of the data may be censored since they are below the reception threshold of the WiFi chipset and that data are dropped at an unknown rate, probably due to limitations of the software. Further, we proposed to cluster the potential user positions in multiple regression classes, where positions in the same regression class share their adaptation parameters, thus gaining flexibility in modelling the deviation of test data from training data. The performance of the algorithms was first validated on artificially generated data and then on real field data of an experimental indoor positioning system. Improvement in positioning result was observed, and the effectiveness of using multiple regression classes was demonstrated.

## 7. REFERENCES


[1] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos, *WLAN Positioning Systems: Principles and Applications in Location-Based Services*, Cambridge University Press, 2012.

[2] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE, 2000, vol. 2, pp. 775–784.

[3] A. Agiwal, P. Khandpur, and H. Saran, "Locator: location estimation system for wireless LANs," in *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hostpots*. ACM, 2004, pp. 102–109.

[4] N. Le Dortz, F. Gain, and P. Zetterberg, "WiFi fingerprint indoor positioning system using probability distribution comparison," in *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 2301–2304.

[5] K. Hoang and R. Haeb-Umbach, "Parameter Estimation and Classification of Censored Gaussian Data with Application to WiFi Indoor Positioning," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, May 2013, IEEE, pp. 3721–3725.

[6] Dimitris Milioris, George Tzagkarakis, Artemis Papakonstantinou, Maria Papadopouli, and Panagiotis Tsakalides, "Low-dimensional signal-strength fingerprint-based positioning in wireless LANs," *Ad Hoc Networks*, 2012.

[7] Gough Lui, Thomas Gallagher, Binghao Li, Andrew G. Dempster, and Chris Rizos, "Differences in RSSI Readings Made by Different Wi-Fi Chipsets: A Limitation of WLAN Localization," in *Proc. International Conference on Localization and GNSS*, Tampere, June 2011, IEEE.

[8] 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 MobiCom04*, Philadelphia, PA, USA, September 2004.

[9] C. J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density HMMs," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[10] M.J.F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[11] M. J. F. Gales and P.C. Woodland, "Mean and variance adaptataion within the MLLR framework," *Computer Speech and Language*, vol. 10, pp. 249–264, 1996.